sgcUI 2026.10

October 2026

Documentación para Delphi

Copyright © 2012-2026 eSeGeCe Software
info@esegece.com
www.esegece.com

Contents

sgcUI 26

sgcUI Overview 26
Instalación 27
Plataformas y paquetes 29
Editions 31
.NET Packages 32

sgcPDF 33

Introducción 33
Instalación 35
Arquitectura 38
Inicio rápido 41
TsgcPDFDocument 44
TsgcPDFEditor 48
TsgcPDFViewer 51
TsgcPDFTextExtractor 53
TsgcPDFExporter 55
TsgcPDFOptimizer 57
TsgcPDFSignatureHandler 59
TsgcPDFSignatureVerifier 61
Texto y fuentes 63
Imágenes 65
Gráficos, transparencia y degradados 66
Tablas 67
Códigos de barras 69
Anotaciones 71
Marcadores 78
Capas 80

Formularios 81
Cifrado 83
Firmas digitales 85
PDF/A 87
Optimización de archivos PDF 89
Combinar, dividir y extraer páginas 90
Extracción de texto 91
Renderizado y exportación 93
Gestión de errores y entradas hostiles 94
Arquitectura del renderizador 96
TsgcPDFRendererRaster 98
TsgcPDFRendererGDI 101
Características de renderizado admitidas 103
TsgcPDFViewerControl 105
TsgcPDFThumbnailView 112
TsgcPDFOutlineView 113
TsgcPDFCommentsView 114
TsgcPDFViewerFMX 116
TsgcPDFCommentsViewFMX 120
Fuentes OpenType 122
Escrituras complejas y shaping 123
Texto en chino, japonés y coreano 125
Extracción de texto: posiciones de caracteres 126
Límites para archivos no confiables 128
Decodificadores de imagen 129
Importación de TIFF 131
TsgcPDFImageExtractor 133
TsgcPDFCanvas 136
EMF a PDF 138
TsgcPDFVCLCanvasHelper 140
HTML a PDF 141

Códigos de barras 2D y GS1-128 144
TsgcPDFBarcodeControl 146
TsgcPDFBarcodeFMX 150
TsgcPDFBarcodeSymbol 153
Linux 158
macOS, iOS y Android 160
C++Builder 161
Archivos adjuntos 162
Facturas Factur-X y ZUGFeRD 165
Guardado incremental y salida compacta 167
Archivos grandes 169
FDF y XFDF 170
Valores y apariencias de formulario 172
Firmas PAdES 174
Sellos de tiempo y validación a largo plazo 176
Verificación de firmas 178
TsgcPDFPAdESSigner (puente sgcSign) 180
Cifrado con certificado 182
TsgcPDFRedactor 184
PDF etiquetado 187
PDF/UA y PDF/A nivel A 190
Validador de accesibilidad 191
Preguntas frecuentes 193

sgcReports 195

Introduction 195
Instalación 196
Getting Started 198
Report Model 200
Expressions and Text Templates 202
Data Sources 204

Groups, Headers, Footers and Columns 205
Fonts 206
Thread Safety and Server Use 207
Accessibility (PDF/UA) 209
PDF/A 210
TsgcReport 211
TsgcReportEngine 212
TsgcReportDataSetSource 214
TsgcReportJSONSource 215
TsgcReportEventSource 216
TsgcReportPDFExport 218
TsgcReportXLSXExport 220
TsgcReportCSVExport 222
TsgcReportImageExport 223
TsgcReportPreviewVCL 224
TsgcReportPreviewFMX 226
TsgcReportHTTPEndpoint 227
TsgcReportHTMLViewer 229
TsgcReportHTMLDesigner 230
TsgcReportBuilder 231
JSON Report Definitions 233
How To: Invoice with Master-Detail Data 235
How To: A Grouped Report with Totals 237
How To: Export to XLSX, CSV and PNG 239
How To: An Accessible PDF Validated with veraPDF 240
How To: Serve Reports Over HTTP 242
How To: A Linux64 Report Daemon 243
How To: A Mobile FMX Report 244
How To: The Web Report Designer 245

sgcCharts 246

Introduction 246
Architecture 247
Editions 249
Instalación 250
Plataformas 252
Quick Start 255
Demos 260
TsgcChart 263
TsgcChartSeries 267
Axes 271
Title, Legend and Margins 275
Themes and Palette 278
Plot Areas 280
Colour Scale 283
TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine 286
TsgcChartSeries_Area 290
TsgcChartSeries_Bar 292
TsgcChartSeries_Scatter 294
TsgcChartSeries_Candlestick, TsgcChartSeries_OHLC 296
TsgcChartSeries_Pie 299
TsgcChartSeries_Radar 303
TsgcChartSeries_Heatmap 306
TsgcChartSeries_Treemap 309
TsgcChartSeries_Funnel 313
TsgcVCLChart 316
TsgcFMXChart 319
TsgcChartRenderer 322
TsgcChartInteraction 325
TsgcChartDataSource 332
TsgcChartDataSource_Manual 334
TsgcChartDataSource_DataSet 336

TsgcChartDataSource_JSON 338
TsgcChartDataSource_CSV 340
TsgcChartDataSource_Table 342
Bridge Overview 346
TsgcChartDataSource_MQTT 347
TsgcChartDataSource_WebSocket 349
TsgcChartDataSource_SSE 351
TsgcChartDataSource_Binance 353
TsgcChartSeries_Indicator 355
TsgcChartSeries_Oscillator 360
TsgcChartSeries_Volume 364
Panes 367
TsgcChartAggregator 371
TsgcChartDrawing 374
TsgcChartBuffer 379
Decimation 382
Canvas 384
Fonts 387
Indicators 390
.NET Overview 393
Inicio Rápido: WinForms 395
Inicio Rápido: WPF 396
Quick Start: Avalonia 397
Headless Rendering 399
Series Gallery 401
Bridge 404
.NET Differences from Delphi 407
Troubleshooting 410

sgcMaps 411

Introducción 411

Descripción general 412
Instalación 415
Inicio rápido VCL 417
Inicio rápido FMX 419
Demos 421
01 Mapa sin conexión 422
02 Características del mapa 423
03 Servicios 425
04 Servicios sin conexión 427
05 Renderizado estático 428
06 FireMonkey 430
07 Teselas vectoriales 431
TsgcVCLMap 432
TsgcFMXMap 437
Teselas, capas y proveedores 441
Teselas vectoriales 444
Marcadores y agrupación 447
Formas y etiquetas 450
Mapas de calor 452
GeoJSON, GPX y KML 454
Funciones de geometría 456
Renderizado de mapas estáticos 458
Teselas sin conexión y el descargador de teselas 461
Lector de OpenStreetMap 464
Índice de geocodificación sin conexión 466
Grafos de rutas y enrutamiento sin conexión 468
TsgcMapService 470
TsgcMapGeocoder 472
TsgcMapDirections 475
TsgcMapPlaces 478
TsgcMapElevation 480

TsgcMapTimeZone 482
TsgcMapRouteCalculator 484
TsgcMapOfflineRouter 487
TsgcMapOfflineGeocoder 489
TsgcMapGeoIndexBuilder 491
TsgcMapLocation 493
Herramientas 496
GeoIndexBuilder 497
RouteGraphBuilder 499
Lazarus 500
Información general de .NET 502
Instalación de .NET 505
Inicio Rápido: WinForms 507
Renderizado sin interfaz visual 509
Datos sin conexión de .NET 511
Correspondencia de clases 514
Diferencias de .NET respecto a Delphi 517
Demos de .NET 519
Solución de problemas 521

sgcDashboard 525

Introducción 525
Arquitectura 526
Ediciones 528
Instalación 529
Plataformas 531
Inicio rápido VCL 534
Inicio rápido FMX 536
Inicio rápido C++Builder 537
Demos 538
TsgcVCLDashboard 539

TsgcVCLDashboardTile 542
TsgcFMXDashboard 544
TsgcFMXDashboardTile 547
TsgcDashboardKPI 549
TsgcVCLKPICard 552
TsgcFMXKPICard 554
TsgcDashboardSource 556
TsgcDashboardSource_WebSocket 558
TsgcDashboardSource_MQTT 560
TsgcDashboardSource_SSE 562
Disposición 564
Temas 566
Persistencia 568
Datos en vivo 570
Imágenes KPI sin interfaz 572
Visión general de .NET 573
Instalación desde NuGet 576
Inicio rápido WinForms 578
Inicio rápido WPF 579
Inicio rápido Avalonia 580
Clases de .NET 581
Diferencias de .NET respecto a Delphi 584
Enlace de datos en .NET 587
Renderizado sin interfaz en .NET 589
Feeds en vivo en .NET 591
Demos de .NET 593
Solución de problemas 595

sgcGrid 597

Introducción 597
Arquitectura 598

Ediciones 600
Instalación 601
Plataformas 603
Rendimiento 606
Inicio rápido VCL 607
Inicio rápido FMX 608
Inicio rápido C++Builder 609
Demos 610
TsgcVCLGrid 611
TsgcFMXGrid 617
TsgcGridColumn 622
TsgcGridColumns 624
TsgcGridData 626
TsgcGridLineView 629
TsgcGridDataStore 633
TsgcGridRenderer 635
TsgcGridFormatRule 638
TsgcGridFormatRules 640
IsgcGridDataHost e IsgcUITable 641
TsgcGridSource_DataSet 643
TsgcGridFeedSource 645
TsgcGridFeedSource_WebSocket 647
TsgcGridFeedSource_MQTT 649
TsgcGridFeedSource_SSE 651
TsgcGridAIQuery 653
Actualizaciones en vivo y subprocesos 655
Ordenación 657
Filtrado 658
Agrupación 659
Resúmenes 662
Filas de árbol 664

Maestro detalle 667
Selección y teclado 671
Cambio de tamaño de columnas 673
Columnas fijas 674
Destello al cambiar 675
Formato condicional 676
Edición en el lugar 678
Modo virtual 680
JSON y CSV 681
Exportación a Excel 682
Enlace con DataSet 685
Flujos en vivo 686
Pregunte a su cuadrícula 688
Gráficos desde la cuadrícula 690
Temas 691
Renderizado sin interfaz 692
Descripción general de .NET 694
Instalación desde NuGet 697
Inicio rápido WinForms 699
Inicio rápido WPF 700
Inicio rápido Avalonia 701
Clases de .NET 702
Diferencias entre .NET y Delphi 706
Enlace de datos en .NET 709
Renderizado sin interfaz en .NET 712
Enlace con DataTable, JSON y CSV en .NET 713
Edición en el lugar en .NET 714
Flujos en vivo en .NET 715
Pregunte a su cuadrícula en .NET 717
Rendimiento en .NET 718
Agrupación y resúmenes de .NET 720

Filas de árbol y maestro detalle de .NET 723
Exportación a Excel de .NET 726
Demos de .NET 728
Solución de problemas 730

sgcNavigation 732

Introducción 732
Arquitectura 734
Ediciones 735
Instalación 736
Plataformas 738
Inicio rápido VCL 741
Inicio rápido FMX 743
Inicio rápido C++Builder 744
Demos 745
TsgcVCLNavigationBar 746
TsgcFMXNavigationBar 749
TsgcNavigationItem 752
TsgcNavigationItems 755
TsgcNavigationModel 757
Modo barra lateral 760
Modo acordeón 762
Contraer y animación 763
Imágenes y glifos 765
Insignias 767
Teclado y foco 768
Vínculos de acción 770
Temas 771
Renderizado sin interfaz 772
Descripción general de .NET 775
Instalación desde NuGet 778

Inicio rápido WinForms 779
Inicio rápido WPF 780
Inicio rápido Avalonia 781
Clases de .NET 782
Diferencias de .NET respecto a Delphi 784
Enlace de datos en .NET 786
Renderizado sin interfaz en .NET 788
Demos de .NET 790
Solución de problemas 792

sgcToolbar 793

Introducción 793
Arquitectura 794
Ediciones 796
Instalación 797
Plataformas 799
Inicio rápido VCL 802
Inicio rápido FMX 804
Inicio rápido C++Builder 805
Demos 806
TsgcVCLToolbar 807
TsgcFMXToolbar 810
TsgcVCLStatusBar 813
TsgcFMXStatusBar 815
TsgcToolbarItem 818
TsgcToolbarItems 821
TsgcStatusBarPanel 822
TsgcStatusBarPanels 824
TsgcToolbarModel 825
TsgcStatusBarModel 828
Tipos de elemento 830

Interruptores y grupos de opciones 832
Menús desplegables 833
Desbordamiento 834
Vínculos de acción 835
Imágenes y glifos 836
Paneles de estado 837
Progreso e indicadores 838
Actualizaciones seguras entre hilos 839
Tirador de tamaño 841
Teclado 842
Temas 843
Renderizado sin interfaz 844
Descripción general de .NET 846
Instalación desde NuGet 849
Inicio rápido de WinForms 850
Inicio rápido de WPF 851
Inicio rápido de Avalonia 852
Clases de .NET 853
Diferencias de .NET con Delphi 856
Enlace de datos en .NET 859
Renderizado sin interfaz en .NET 861
Demos de .NET 863
Solución de problemas 865

sgcWizard 867

Introducción 867
Arquitectura 868
Ediciones 870
Instalación 871
Plataformas 873
Inicio rápido VCL 876

Inicio rápido FMX ... 878
Inicio rápido C++Builder ... 880
Demos ... 881
TsgcVCLWizard ... 882
TsgcVCLWizardPage ... 885
TsgcFMXWizard ... 887
TsgcFMXWizardPage ... 890
TsgcVCLStepper ... 892
TsgcFMXStepper ... 894
TsgcWizardStep ... 896
TsgcWizardSteps ... 898
TsgcStepperModel ... 899
Navegación y páginas omitidas ... 901
Validación ... 903
Finalizar y cancelar ... 904
Teclado ... 905
Estados de los pasos ... 906
Disposición del stepper ... 908
Títulos de los botones y localización ... 910
Temas ... 911
Tiempo de diseño ... 912
Renderizado sin interfaz ... 913
Descripción general de .NET ... 915
Instalación desde NuGet ... 917
Inicio rápido WinForms ... 918
Inicio rápido WPF ... 919
Inicio rápido Avalonia ... 920
Clases de .NET ... 921
Diferencias de .NET respecto a Delphi ... 924
Enlace de datos en .NET ... 926
Renderizado sin interfaz en .NET ... 928

Demos de .NET 930
Solución de problemas 932

sgcPivotGrid 933

Introducción 933
Arquitectura 935
Ediciones 937
Instalación 938
Plataformas 940
Inicio rápido VCL 943
Inicio rápido FMX 945
Inicio rápido C++Builder 947
Inicio rápido DataSet 948
Inicio rápido sin interfaz 950
Demos 952
TsgcVCLPivotGrid 953
TsgcFMXPivotGrid 957
TsgcVCLPivotFieldList 960
TsgcFMXPivotFieldList 962
TsgcPivotEngine 964
TsgcPivotField 968
TsgcPivotCalculatedField 970
TsgcPivotSource_DataSet 971
TsgcPivotSource_Table 972
TsgcPivotGridRenderer 974
TsgcPivotFormatRule 976
Áreas 978
Agregaciones 980
Agrupación de fechas 982
Totales 984
Filtros 986

Top N 989
Mostrar valores como 990
Campos calculados 992
Persistencia del diseño 994
Exportación 996
Expandir y contraer 998
Ordenación 1000
Drill-down 1002
Formato condicional 1004
Interoperabilidad con gráficos 1006
Teclado y accesibilidad 1008
Rendimiento 1010
TsgcPivotGridAIQuery 1012
Pregunte a su tabla dinámica 1016
Descripción general de .NET 1021
Instalación desde NuGet 1023
Inicio rápido WinForms 1024
Inicio rápido WPF 1025
Inicio rápido Avalonia 1026
Clases de .NET 1027
Diferencias de .NET respecto a Delphi 1030
Enlace de datos en .NET 1032
Renderizado sin interfaz en .NET 1034
Demos de .NET 1035
Solución de problemas 1036

sgcPlanner 1038

Introducción 1038
Arquitectura 1039
Ediciones 1042
Instalación 1044

Plataformas 1046
Inicio rápido VCL 1050
Inicio rápido FMX 1052
Inicio rápido C++Builder 1053
Demos 1054
TsgcPlannerData 1056
TsgcVCLScheduler 1061
TsgcFMXScheduler 1065
TsgcVCLGantt 1068
TsgcFMXGantt 1072
TsgcVCLKanban 1076
TsgcFMXKanban 1080
TsgcPlannerSource_DataSet 1083
TsgcPlannerICalendar 1086
TsgcPlannerMSProject 1088
TsgcPlannerRenderer 1090
Datos compartidos 1093
Vistas sincronizadas 1095
Calendario laboral 1097
Recurrencia 1099
Escala de tiempo y zoom 1102
Arrastrar y soltar 1104
Dependencies 1107
Programación automática 1109
Ruta crítica 1111
Líneas base 1113
Carriles y límites de WIP 1115
Vinculación con DataSet 1117
Persistencia JSON 1119
iCalendar 1121
XML de MS Project 1123

Renderizado sin ventana 1125
Temas 1127
Características por edición 1129
Sincronización con Google Calendar 1131
TsgcPlannerGoogleSync 1135
TsgcPlannerICalSubscription 1140
Descripción general de .NET 1144
Instalación desde NuGet 1147
Inicio rápido WinForms 1148
Inicio rápido WPF 1149
Inicio rápido Avalonia 1151
Clases de .NET 1152
Diferencias de .NET respecto a Delphi 1155
Datos y programación en .NET 1157
Vinculación de datos en .NET 1158
JSON, iCalendar y MS Project en .NET 1160
Renderizado sin ventana en .NET 1161
Sincronización con Google Calendar e iCalendar en .NET 1162
Demos de .NET 1164
Solución de problemas 1166

sgcEditors 1169

Introducción 1169
Arquitectura 1171
Controles 1174
Instalación 1176
Plataformas 1178
Inicio rápido VCL 1181
Inicio rápido FMX 1183
Inicio rápido C++Builder 1185
Inicio rápido data-aware 1186

Calendario mensual 1188
Selector de fecha 1192
Selector de hora 1196
Selector de fecha y hora 1199
Control deslizante de rango 1202
Selector de color 1205
Token edit 1208
Valoración 1211
Spin edit 1214
Selector de fecha DB 1217
Selector de hora DB 1219
Spin edit DB 1221
Valoración DB 1223
Control deslizante de rango DB 1225
TsgcEditorModel 1227
TsgcEditorLocale 1231
TsgcEditorFieldLink 1234
Entrada de texto e IME 1237
Teclado y accesibilidad 1240
Temas 1242
Localización 1244
DPI 1246
Salida sin ventana 1248
Ediciones 1250
Descripción general de .NET 1252
Instalación desde NuGet 1254
Inicio rápido WinForms 1255
Inicio rápido WPF 1256
Inicio rápido Avalonia 1257
Clases de .NET 1258
Diferencias de .NET respecto a Delphi 1261

Enlace de datos en .NET 1263
Renderizado sin ventana en .NET 1264
Demos de .NET 1265
Solución de problemas 1266

sgcGauges 1268

Introducción 1268
Arquitectura 1269
Tipos de medidor 1271
Instalación 1273
Plataformas 1275
Inicio rápido VCL 1278
Inicio rápido FMX 1280
Inicio rápido sin ventana 1282
Inicio rápido C++Builder 1283
Medidor radial 1284
Medidor lineal 1286
Medidor digital 1288
LED 1290
Barra de LEDs 1292
Anillo de progreso 1294
VU meter 1296
Reloj 1298
Brújula 1300
Anunciador 1302
TsgcCustomGauge 1304
TsgcGaugeScale y TsgcGaugeRange 1307
TsgcGaugePointer, TsgcGaugeThreshold y TsgcGaugeAnimation 1309
Escalas 1312
Rangos 1314
Punteros 1316

Umbrales 1318
Animación 1320
Temas 1322
Salida sin ventana 1324
Ediciones 1326
Descripción general del puente 1328
Fuente MQTT 1331
Fuente WebSocket 1333
Fuente de eventos enviados por el servidor 1335
Descripción general de .NET 1337
Instalación desde NuGet 1340
Inicio rápido WinForms 1341
Inicio rápido WPF 1342
Inicio rápido Avalonia 1343
Clases de .NET 1344
Diferencias de .NET respecto a Delphi 1347
Enlace de datos en .NET 1348
Renderizado sin ventana en .NET 1349
Puente sgcWebSockets en .NET 1350
Demos de .NET 1352
Solución de problemas 1353

sgcDiagram 1355

Introducción 1355
Qué es sgcDiagram 1357
Arquitectura 1358
Ediciones 1360
Plataformas 1361
Inicio rápido VCL 1365
Inicio rápido FMX 1367
Inicio rápido C++Builder 1368

Inicio rápido sin ventana 1369
TsgcVCLDiagramView 1370
TsgcFMXDiagramView 1373
TsgcVCLDiagramPalette 1374
TsgcFMXDiagramPalette 1376
TsgcDiagramDataBinding 1377
Eventos 1379
TsgcDiagramDocument 1381
TsgcDiagramNode 1384
TsgcDiagramConnector 1386
TsgcDiagramPort 1388
TsgcDiagramStyle 1389
TsgcDiagramLayer 1390
TsgcDiagramController 1391
TsgcDiagramRenderer 1393
Formato JSON 1394
Diseño por capas 1397
Diseño en árbol 1399
Enlace de datos 1401
Formas 1403
Conectores y enrutamiento 1405
Puertos y anclaje 1407
Edición y teclado 1409
Ajuste y guías 1411
Deshacer y portapapeles 1412
Capas 1413
Serialización 1414
Exportación 1416
Temas 1417
Modo visor 1418
Descripción general de .NET 1419

Instalación desde NuGet 1422
Inicio rápido WinForms 1423
Inicio rápido WPF 1424
Inicio rápido Avalonia 1425
Clases de .NET 1426
Diferencias de .NET respecto a Delphi 1428
Enlace de datos en .NET 1430
Renderizado sin ventana en .NET 1432
Demos de .NET 1433
Solución de problemas 1434

Index 1435

Información general de sgcUI

sgcUI es la suite de componentes visuales y de documentos de eSeGeCe para Delphi y C++Builder. Los productos comparten una carpeta de código fuente, un archivo de inclusión de versión, un instalador y esta ayuda. Cada producto se licencia por separado, y el instalador instala solo los productos que cubre su licencia.

Los productos

Producto	Para qué sirve
sgcPDF	Una biblioteca PDF nativa para crear, editar y renderizar documentos PDF, con texto y fuentes, dibujo, imágenes, seguridad, accesibilidad y controles de visor. Véase sgcPDF.
sgcReports	Un motor de informes sin interfaz gráfica basado en sgcPDF, para servicios de Windows, daemons de Linux64 y endpoints HTTP, además de aplicaciones VCL y FMX. Véase sgcReports.
sgcCharts	Gráficos de datos en vivo. Las fuentes de datos añaden puntos o velas OHLC a un búfer circular, y el gráfico dibuja una instantánea diezmada, de modo que incluso los flujos rápidos se renderizan con fluidez. Véase sgcCharts.
sgcMaps	Mapas interactivos. Los controles de mapa dibujan teselas ráster y vectoriales de forma nativa, sin control de navegador, y muestran sobre ellas marcadores, rutas, formas, etiquetas y mapas de calor. Véase sgcMaps.
sgcDashboard	Un dashboard formado por una cuadrícula de tiles. Cada tile aloja cualquier control, como un gráfico, un mapa, un visor de PDF o una vista previa de informe. Véase sgcDashboard.
sgcGrid	Una cuadrícula de datos en vivo. Las filas se actualizan por clave desde cualquier hilo, las celdas modificadas destellan y se desvanecen, y las ráfagas de actualizaciones se agrupan en repintados temporizados. Véase sgcGrid.
sgcNavigation	Una barra de navegación que muestra las páginas de una aplicación como una lista de elementos con iconos, títulos, insignias y encabezados de grupo. Véase sgcNavigation.
sgcToolbar	Una barra de herramientas y una barra de estado, con botones, interruptores, grupos de opciones, botones desplegables, etiquetas, separadores y espaciadores. Véase sgcToolbar.
sgcWizard	Asistentes paso a paso e indicadores de pasos: contenedores de páginas con una banda de stepper y una barra de botones. Véase sgcWizard.
sgcUICore	La base compartida de sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard. No se vende ni se selecciona por separado: el instalador la añade siempre que se instala uno de esos productos.

Siguientes pasos

- Instalación explica el instalador de sgcUI y cómo elige los productos.
- Ediciones enumera las ediciones Registered, Trial y Community de cada producto.
- Paquetes .NET enumera los paquetes NuGet de cada producto.

Instalación

Todos los productos de sgcUI se instalan con un único instalador. No es necesario descargar ni ejecutar un instalador distinto para cada producto.

Inicio de sesión

El instalador Registered le pide que inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe. También puede reutilizar la licencia almacenada por una instalación anterior o activar una licencia sin conexión. El servidor de licencias responde con los productos que posee su cuenta, y una licencia All-Access incluye todos los productos.

Elección de los productos

La página de componentes muestra una casilla por producto. Un producto que su licencia no cubre aparece atenuado y sin marcar. Seleccione los productos que desea instalar, junto con las demos y los archivos de ayuda. Algunos productos necesitan otros, y el instalador los añade por usted:

- Al seleccionar sgcReports también se instala sgcPDF, el motor en el que se basa.
- sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard se basan en sgcUICore, por lo que sgcUICore se instala siempre que se selecciona uno de ellos.

Integraciones con sgcWebSockets

El grupo **sgcWebSockets bridges** ofrece sgcCharts for sgcWebSockets, sgcDashboard for sgcWebSockets, sgcGrid for sgcWebSockets y sgcReports HTML for sgcWebSockets. Un puente solo está disponible cuando sgcWebSockets ya está instalado en el equipo y usted posee el producto que amplía. De lo contrario aparece atenuado.

Compilación de los paquetes

Tras copiar los archivos, el instalador compila los paquetes en su equipo para cada versión de Delphi y de C++Builder que haya elegido y registra los paquetes de tiempo de diseño en el IDE. Los paquetes se compilan en orden de dependencia: primero sgcUICore, después los productos basados en él, luego sgcPDF, sgcReports y sgcMaps, y por último los puentes de sgcWebSockets.

Migración desde instalaciones independientes

Las versiones anteriores de sgcPDF, sgcReports, sgcMaps y sgcCharts tenían instaladores propios. Cuando el instalador de sgcUI encuentra una de esas instalaciones, enumera las rutas de biblioteca y los paquetes de diseño que la instalación antigua añadió a cada IDE, y solo los elimina después de que usted confirme esa página. También se eliminan las entradas de las rutas de búsqueda y de exploración que aún apuntan a la carpeta de un producto independiente, de modo que el IDE nunca encuentra dos copias de una unidad.

Ediciones Trial y Community

- El instalador Trial instala todos los productos en su edición Trial, sin licencia. Hay un instalador Trial para Delphi y otro para C++Builder.
- El instalador Community instala sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard, junto con sgcUICore. También ofrece sgcReports cuando ya hay un sgcPDF con licencia instalado, porque sgcPDF no tiene edición Community. El instalador Community nunca ofrece los puentes de sgcWebSockets. Véase Ediciones.

Detalles de los productos

Cada producto describe sus paquetes, requisitos y archivos instalados en su propio tema de Instalación:

- sgcPDF
- sgcReports
- sgcCharts
- sgcMaps
- sgcDashboard
- sgcGrid
- sgcNavigation
- sgcToolbar
- sgcWizard

Plataformas y paquetes

sgcUICore y cada producto de componentes de sgcUI incluyen un paquete de runtime independiente del framework (sgc<Product>), un paquete de runtime VCL (sgc<Product>VCL), un paquete de runtime FireMonkey desde XE6 (sgc<Product>FMX), y un paquete de tiempo de diseño VCL y otro FireMonkey (dclsgc<Product> y dclsgc<Product>FMX). sgcUICore no tiene paquete de tiempo de diseño. El paquete neutral se compila para Win32 y Win64, y desde Delphi 10.3 también para Linux64 y macOS. El paquete FireMonkey se compila para macOS desde Delphi 10.3 y para Android e iOS desde Delphi 10.4. Los paquetes de C++Builder se compilan solo para Windows. Cada producto enumera sus paquetes, versiones del IDE y plataformas de destino en su propio tema:

Producto	Tema
sgcCharts	Plataformas
sgcDashboard	Plataformas
sgcGrid	Plataformas
sgcNavigation	Plataformas
sgcToolbar	Plataformas
sgcWizard	Plataformas
sgcPivotGrid	Plataformas
sgcPlanner	Plataformas
sgcEditors	Plataformas
sgcGauges	Plataformas
sgcDiagram	Plataformas

C++Builder Win64: paquetes de runtime

C++Builder Win64 (cadena de herramientas clásica, C++Builder 12 y 13): los paquetes de runtime Win64 de sgcUI los enlaza el compilador de Delphi, por lo que las excepciones gestionadas dentro del código de sgcUI funcionan cuando su aplicación usa paquetes de runtime. El enlazador C++ Win64 (ilink64) tiene todavía un defecto propio: el código Delphi que enlaza estáticamente en una aplicación compilada con paquetes de runtime se bloquea en `_DelphiExceptionHandler` la primera vez que un bloque `except on E: <class>` captura una excepción cuya clase está en un paquete de runtime.

- Cuando Enlazar con paquetes de runtime está activado (Opciones del proyecto, Paquetes, Paquetes de runtime), enumere todos los paquetes de runtime de sgcUI que usa su proyecto, para que ninguna unidad de sgcUI se enlace en el ejecutable.
- En el .cbproj, cada uno de esos paquetes debe estar tanto en PackageImports como en LinkPackageImports. Un paquete que solo está en PackageImports no se enlaza.
- Cuando Enlazar con paquetes de runtime está desactivado, todo se enlaza de forma estática y funciona.
- Win32 y Win64x (Modern) no se ven afectados.

C++Builder 10.3 Rio, Win64: excepciones en hilos secundarios

Cuando se lanza una excepción de Delphi en un hilo secundario, por ejemplo dentro de un manejador de eventos que se ejecuta en un hilo de sgcWebSockets o en cualquier TThread, y el código C++ de ese hilo la captura, o se desenrolla a través de una función C++ que destruye un objeto local como un UnicodeString temporal, la aplica-

ción se bloquea con una violación de acceso. Es un defecto del runtime de C++Builder 10.3 Win64 (cc64260mt.dll). También ocurre sin código de sgc, y C++Builder 10.1, 10.2, 10.4, 11, 12 y 13 no se ven afectados. Soluciones: use C++Builder 10.4 o posterior, o en el código Win64 que se ejecuta en hilos secundarios evite lanzar excepciones, por ejemplo TryStrToInt en lugar de StrToInt.

C++Builder 13 Win64x: nombres de archivo de los paquetes

Los paquetes de runtime Win64x (Modern) tienen los mismos nombres de archivo que los paquetes Win64 clásicos, por ejemplo sgcUICoreC13.bpl, y ambos son de 64 bits, por lo que Windows no puede distinguirlos. Una aplicación Win64x que usa paquetes de runtime y encuentra primero el bpl de Win64 no arranca (0xC0000139, punto de entrada no encontrado). Ponga la carpeta de los bpl Win64x (Bpl\Win64x) en el PATH antes de la carpeta Win64, o distribuya los bpl Win64x en la misma carpeta que el ejecutable. Las aplicaciones enlazadas sin paquetes de runtime no se ven afectadas.

Delphi XE7 FireMonkey: FMX.Forms primero

Solo Delphi XE7: un programa FireMonkey puede terminar con "Runtime error 216" al cerrarse cuando FMX.Types o una unidad FMX de sgcUI va antes de FMX.Forms en el orden de uses del programa y se usó un control con estilo (FMX libera el servicio de selección antes que el estilo predeterminado). XE6, XE8 y las versiones posteriores no se ven afectados. Solución: ponga FMX.Forms como primera unidad en la cláusula uses del .dpr, como hace el IDE en las nuevas aplicaciones FMX.

Capturar un formulario VCL con PaintTo

Para capturar un formulario VCL que contiene controles de sgcUI en un mapa de bits con PaintTo, bloquee el canvas del mapa de bits alrededor de la llamada (`Bitmap.Canvas.Lock`, y después `Unlock`). PaintTo sobre un canvas sin bloquear puede perder su contexto de dispositivo cuando un control nativo del formulario procesa un mensaje durante la captura, y el mapa de bits queda en blanco o incompleto. Un formulario FireMonkey usa MakeScreenshot en su lugar.

Véase también

- Instalación
- Editions

Ediciones

Cada producto de sgcUI se compila a partir del mismo código fuente en todas las ediciones. La edición solo decide qué unidades y funciones incluye una compilación.

- **Registered**: todas las unidades de los productos que incluye su licencia, incluidos los puentes de sgcWebSockets.
- **Trial**: todos los productos, para evaluación. Las compilaciones Trial se generan con el define `SGC_TRIAL`.
- **Community**: una edición gratuita para los productos que se enumeran a continuación, compilada con el define `SGC_COMMUNITY`. Nunca incluye los puentes de sgcWebSockets.

Edición Community por producto

Producto	Community	No incluido en Community
sgcPDF	No	El instalador Community no incluye sgcPDF.
sgcReports	Sí, sobre un sgcPDF con licencia	El puente HTML de sgcReports para sgcWebSockets. sgcReports Community solo se ofrece cuando ya hay un sgcPDF con licencia instalado.
sgcCharts	Sí	Las funciones financieras: indicadores, series financieras, el agregador y las herramientas de dibujo. El puente de sgcWebSockets.
sgcMaps	No	El instalador Community no incluye sgcMaps.
sgcDashboard	Sí	Arrastrar y redimensionar tiles en tiempo de ejecución, y SaveLayout y LoadLayout. El puente de sgcWebSockets.
sgcGrid	Sí	Edición de celdas en el lugar. El puente de sgcWebSockets.
sgcNavigation	Sí	Los enlaces de Action en los elementos de navegación.
sgcToolbar	Sí	El menú de desbordamiento y los enlaces de Action.
sgcWizard	Sí	Nada. La compilación Community tiene el conjunto completo de funciones.
sgcUICore	Sí	Se instala con los productos Community que lo necesitan.

Cada producto explica sus ediciones con más detalle en su propia ayuda, por ejemplo Ediciones de sgcCharts.

Paquetes .NET

Las versiones .NET de los productos de sgcUI se distribuyen como paquetes NuGet. Cada producto tiene un paquete principal, con el nombre del producto, y paquetes complementarios para un framework de interfaz de usuario o una integración. Agregue el paquete principal y los complementarios que necesite su aplicación.

Producto	Paquete principal	Paquetes complementarios
sgcPDF	`esegece.sgcPDF`	`esegece.sgcPDF.WinForms`, `esegece.sgcPDF.sgcSign`
sgcReports	`esegece.sgcReports`	`esegece.sgcReports.WinForms`, `esegece.sgcReports.AspNetCore`, `esegece.sgcReports.HTML`, `esegece.sgcReports.sgcSign`
sgcCharts	`esegece.sgcCharts`	`esegece.sgcCharts.Rendering`, `esegece.sgcCharts.WinForms`, `esegece.sgcCharts.Wpf`, `esegece.sgcCharts.Avalonia`, `esegece.sgcCharts.sgcWebSockets`
sgcMaps	`esegece.sgcMaps`	`esegece.sgcMaps.Rendering`, `esegece.sgcMaps.WinForms`, `esegece.sgcMaps.Offline`
sgcDashboard	`esegece.sgcDashboard`	`esegece.sgcDashboard.Rendering`, `esegece.sgcDashboard.WinForms`, `esegece.sgcDashboard.Wpf`, `esegece.sgcDashboard.Avalonia`, `esegece.sgcDashboard.sgcWebSockets`
sgcGrid	`esegece.sgcGrid`	`esegece.sgcGrid.Rendering`, `esegece.sgcGrid.WinForms`, `esegece.sgcGrid.Wpf`, `esegece.sgcGrid.Avalonia`, `esegece.sgcGrid.sgcWebSockets`, `esegece.sgcGrid.sgcHTML`
sgcNavigation	`esegece.sgcNavigation`	`esegece.sgcNavigation.Rendering`, `esegece.sgcNavigation.WinForms`, `esegece.sgcNavigation.Wpf`, `esegece.sgcNavigation.Avalonia`
sgcToolbar	`esegece.sgcToolbar`	`esegece.sgcToolbar.Rendering`, `esegece.sgcToolbar.WinForms`, `esegece.sgcToolbar.Wpf`, `esegece.sgcToolbar.Avalonia`
sgcWizard	`esegece.sgcWizard`	`esegece.sgcWizard.Rendering`, `esegece.sgcWizard.WinForms`, `esegece.sgcWizard.Wpf`, `esegece.sgcWizard.Avalonia`
sgcUICore	`esegece.sgcUICore`	`esegece.sgcUICore.Rendering`, `esegece.sgcUICore.WinForms`, `esegece.sgcUICore.Wpf`, `esegece.sgcUICore.Avalonia`. Estos paquetes son dependencias de los productos basados en sgcUICore, por lo que NuGet los restaura automáticamente.

Paquetes Community

Una compilación Community de un paquete tiene el mismo id con el sufijo `.Community`, por ejemplo `esegece.sgcGrid.Community`. Consulte Ediciones para saber qué deja fuera la edición Community de cada producto.

Introducción

sgcPDF es una biblioteca de Delphi para crear, abrir, editar, firmar, renderizar y ver documentos PDF. Está escrita en Object Pascal puro y se compila dentro de su ejecutable, por lo que no hay que distribuir ninguna DLL del motor PDF ni licenciar ninguna biblioteca de terceros.

Características principales

- Cree documentos PDF desde código: texto con las 14 fuentes estándar o fuentes TrueType y OpenType (CFF) incrustadas con subconjuntos (subsetting), texto Unicode incluido CJK, composición de escritura compleja (unión de caracteres árabes, ligaduras, texto de derecha a izquierda), líneas, curvas, imágenes, transparencia, degradados y patrones.
- Dibujo con una API similar a TCanvas (TsgcPDFCanvas), dibujo de canvas EMF y VCL convertido a PDF vectorial, y HTML con CSS convertido a páginas PDF.
- Imágenes: importación de JPEG, PNG, BMP y TIFF multipágina; decodificadores en Pascal puro para JPEG (baseline y progresivo), JPEG 2000, CCITT G3/G4 y JBIG2; extracción de imágenes a PNG, JPEG o BMP.
- Tablas con encabezados, bordes, colores de fila alternos y saltos de página automáticos.
- Códigos de barras: Code 128, GS1-128, Code 39, EAN-13, EAN-8, UPC-A, ITF, Codabar, QR Code (versiones 1 a 40), DataMatrix y PDF417.
- Anotaciones, marcadores (esquemas), capas de contenido opcional y formularios interactivos: creación, relleno, aplanado, listas de selección múltiple, apariencias de campo Unicode, importación y exportación de FDF y XFDF.
- Abra documentos existentes, edítelos, agregue marcas de agua, encabezados, pies de página y números de página, combine, divida, extraiga, rote y elimine páginas.
- Guardado incremental que mantiene válidas las firmas, salida compacta con object streams y cross-reference streams, y lectura bajo demanda de archivos de más de 2 GB.
- Archivos incrustados con relaciones PDF/A-3, y facturas electrónicas Factur-X, ZUGFeRD y XRechnung.
- Extracción de texto con la posición, la fuente y el tamaño de cada carácter.
- Cifrado con contraseñas (RC4 de 40 y 128 bits, AES-128, AES-256 revisión 6) o con certificados de destinatario (cifrado de clave pública).
- Firmas digitales con archivos PFX, el almacén de certificados de Windows o cualquier firmante externo, PAdES (ETSI.CAdES.detached), sellos de tiempo de documento, DSS para validación a largo plazo, verificación de firmas, y PAdES de B-B a B-LTA mediante el puente opcional sgcSign.
- Redacción real que elimina texto, píxeles de imagen, gráficos y anotaciones de las áreas marcadas o de los resultados de búsqueda.
- PDF etiquetado con una API de árbol de estructura, salida PDF/UA-1 y PDF/A-1a, 2a y 3a, salida PDF/A-1b, 2b y 3b, y un validador de PDF/A y PDF/UA.
- Optimizador que comprime streams, elimina objetos duplicados y no utilizados, y reduce la resolución de las imágenes.
- Renderizado en Pascal puro de páginas a PNG y BMP en todas las plataformas, un renderizador GDI, y controles de visor para VCL y FireMonkey con zoom, selección, búsqueda, relleno de formularios e impresión.
- Windows, Linux de 64 bits, macOS, iOS y Android, con Delphi 7 a 13 y C++Builder 2007 a 13.

Requisitos

- Delphi 7 a Delphi 13, C++Builder 2007 a C++Builder 13.
- Windows de 32 y 64 bits para todas las funciones.
- Linux de 64 bits, macOS, iOS y Android para el núcleo multiplataforma, el renderizador ráster y el visor FireMonkey (Delphi XE5 o posterior para FireMonkey).
- El renderizador GDI, TsgcPDFViewer, TsgcPDFExporter y los controles de visor VCL utilizan GDI de Windows. La firma con archivos PFX o el almacén de certificados de Windows y el cifrado con certificados utilizan CryptoAPI de Windows (crypt32.dll, parte de Windows).

Componentes

Después de la instalación, los siguientes componentes aparecen en la página **sgcPDF** de la paleta de herramientas:

Componente	Paquete	Descripción
TsgcPDFDocument	Core	El documento PDF: crea páginas, dibuja contenido, carga, guarda, cifra.
TsgcPDFEditor	Core	Edición de archivos PDF existentes: marcas de agua, números de página, combinación, formularios.
TsgcPDFTextExtractor	Core	Texto de una página con las fuentes y las posiciones de los caracteres.
TsgcPDFOptimizer	Core	Reduce el tamaño de un archivo PDF.
TsgcPDFSignatureHandler	Core	Firma documentos PDF.
TsgcPDFSignatureVerifier	Core	Verifica las firmas de un documento PDF.
TsgcPDFRedactor	Core	Elimina texto, imágenes y gráficos de áreas de las páginas.
TsgcPDFImageExtractor	Core	Enumera y guarda las imágenes de un PDF.
TsgcPDFHTMLRenderer	Core	Convierte HTML y CSS en páginas PDF.
TsgcPDFViewer	VCL	Una ventana nativa que muestra, busca e imprime un PDF.
TsgcPDFExporter	VCL	Renderiza páginas a archivos BMP, PNG o JPEG y miniaturas.
TsgcPDFViewerControl	VCL	Control de visor VCL: zoom, selección, búsqueda, relleno de formularios, impresión.
TsgcPDFThumbnailView	VCL	Miniaturas de páginas para TsgcPDFViewerControl.
TsgcPDFOutlineView	VCL	Árbol de marcadores para TsgcPDFViewerControl.
TsgcPDFBarcodeControl	VCL	Control de código de barras y QR Code: once simbologías, giro, salida PNG, SVG y PDF.
TsgcPDFViewerFMX	FMX	Control de visor FireMonkey para Windows, macOS, iOS, Android y Linux.
TsgcPDFBarcodeFMX	FMX	Control FireMonkey de código de barras y QR Code.

Comience con Instalación y después con Inicio rápido.

Instalación

Requisitos

- Delphi 7 a Delphi 13, o C++Builder 2007 a C++Builder 13.
- Windows de 32 y 64 bits para todas las funciones. Linux de 64 bits, macOS, iOS y Android para las unidades multiplataforma (Delphi XE5 o posterior para FireMonkey), vea Linux y macOS, iOS y Android.

Paquetes

sgcPDF está dividido en tres paquetes de tiempo de ejecución, de modo que las aplicaciones y servicios que no utilizan la VCL o FireMonkey solo necesitan el núcleo:

Paquete	Requiere	Contenido
sgcPDFDxx (núcleo, tiempo de ejecución)	rtl	Todas las unidades independientes de la plataforma: documento, editor, analizador, escritor, fuentes, shaping, imágenes y decodificadores, formularios, anotaciones, cifrado, firmas, PDF/A, PDF etiquetado, redacción, adjuntos, extracción de texto e imágenes, optimizador, TsgcPDFCanvas, HTML a PDF y el renderizador ráster.
sgcPDFVCLDxx (VCL, tiempo de ejecución)	rtl, vcl, sgcPDFDxx	Unidades de Windows y VCL: TsgcPDFRendererGDI, TsgcPDFViewer, TsgcPDFExporter, TsgcPDFViewerControl, TsgcPDFThumbnailView, TsgcPDFOutlineView, TsgcPDFBarcodeControl, TsgcPDFEMFPlayer y TsgcPDFVCLCanvasHelper.
sgcPDFFMXDxx (FMX, tiempo de ejecución, XE5 y posterior)	rtl, fmx, sgcPDFDxx	TsgcPDFViewerFMX, TsgcPDFBarcodeFMX y TsgcPDFRendererFMX.
dclsgcPDFDxx (tiempo de diseño)	designide, rtl, vcl, sgcPDFDxx, sgcPDFVCLDxx	Registra los componentes del núcleo y de VCL.
dclsgcPDFFMXDxx (tiempo de diseño, XE5 y posterior)	designide, rtl, fmx, sgcPDFDxx, sgcPDFFMXDxx	Registra TsgcPDFViewerFMX y TsgcPDFBarcodeFMX.

xx es el sufijo de la versión de Delphi, y los archivos están en la carpeta **Packages**:

Versión de Delphi	Sufijo	Paquetes FMX
Delphi 7	D7	No
Delphi 2007	D2007	No
Delphi 2009	D2009	No
Delphi 2010	D2010	No
Delphi XE a XE4	DXE ... DXE4	No
Delphi XE5 a XE8	DXE5 ... DXE8	Sí

Versión de Delphi	Sufijo	Paquetes FMX
Delphi 10 Seattle	D10	Sí
Delphi 10.1 Berlin	D10_1	Sí
Delphi 10.2 Tokyo	D10_2	Sí
Delphi 10.3 Rio	D10_3	Sí
Delphi 10.4 Sydney	D10_4	Sí
Delphi 11 Alexandria	D11	Sí
Delphi 12 Athens	D12	Sí
Delphi 13	D13	Sí

Por ejemplo, Delphi 13 utiliza sgcPDFD13.dpk, sgcPDFVCLD13.dpk, sgcPDFFMXD13.dpk, dclsgcPDFD13.dpk y dclsgcPDFFMXD13.dpk. C++Builder utiliza los paquetes .cbproj sgcPDFCxx, sgcPDFVCLCxx y dclsgcPDFCxx, vea C++Builder.

Instalación de los paquetes

1. Abra y compile (o construya) los paquetes de tiempo de ejecución en este orden: **sgcPDFDxx.dpk**, **sgcPDFVCLDxx.dpk** y, si utiliza FireMonkey, **sgcPDFFMXDxx.dpk**.
2. Abra el paquete de tiempo de diseño **dclsgcPDFDxx.dpk**, compílelo y seleccione **Install**. Para FireMonkey, haga lo mismo con **dclsgcPDFFMXDxx.dpk**.
3. Agregue la carpeta **Source** de sgcPDF a la ruta de la biblioteca: *Tools* > *Options* > *Language* > *Delphi* > *Library* > *Library Path* (en versiones anteriores *Tools* > *Options* > *Library*). Agréguela para cada plataforma que utilice: Windows de 32 bits, Windows de 64 bits, Linux de 64 bits, macOS, iOS y Android.

Verificación de la instalación

Cree una nueva aplicación VCL y abra la paleta de herramientas. La página **sgcPDF** contiene estos componentes (TsgcPDFViewerFMX y TsgcPDFBarcodeFMX aparecen en las aplicaciones FireMonkey):

Componente	Paquete	Descripción
TsgcPDFDocument	Core	El documento PDF: crea páginas, dibuja contenido, carga, guarda, cifra.
TsgcPDFEditor	Core	Edición de archivos PDF existentes: marcas de agua, números de página, combinación, formularios.
TsgcPDFTextExtractor	Core	Texto de una página con las fuentes y las posiciones de los caracteres.
TsgcPDFOptimizer	Core	Reduce el tamaño de un archivo PDF.
TsgcPDFSignatureHandler	Core	Firma documentos PDF.
TsgcPDFSignatureVerifier	Core	Verifica las firmas de un documento PDF.
TsgcPDFRedactor	Core	Elimina texto, imágenes y gráficos de áreas de las páginas.

Componente	Paquete	Descripción
TsgcPDFImageExtractor	Core	Enumera y guarda las imágenes de un PDF.
TsgcPDFHTMLRenderer	Core	Convierte HTML y CSS en páginas PDF.
TsgcPDFViewer	VCL	Una ventana nativa que muestra, busca e imprime un PDF.
TsgcPDFExporter	VCL	Renderiza páginas a archivos BMP, PNG o JPEG y miniaturas.
TsgcPDFViewerControl	VCL	Control de visor VCL: zoom, selección, búsqueda, relleno de formularios, impresión.
TsgcPDFThumbnailView	VCL	Miniaturas de páginas para TsgcPDFViewerControl.
TsgcPDFOutlineView	VCL	Árbol de marcadores para TsgcPDFViewerControl.
TsgcPDFBarcodeControl	VCL	Control de código de barras y QR Code: once simbologías, giro, salida PNG, SVG y PDF.
TsgcPDFViewerFMX	FMX	Control de visor FireMonkey para Windows, macOS, iOS, Android y Linux.
TsgcPDFBarcodeFMX	FMX	Control FireMonkey de código de barras y QR Code.

Arrastre un componente al formulario y compile el proyecto.

Usar sgcPDF sin instalarlo

Todos los componentes también se pueden crear en tiempo de ejecución. Cada componente tiene un constructor que no necesita un formulario, por ejemplo `TsgcPDFDocument.Create`, de modo que las aplicaciones de consola, los servicios, los servidores web y las DLL solo necesitan la carpeta **Source** en la ruta de búsqueda. La mayoría de los ejemplos de esta ayuda son programas de consola.

Arquitectura

sgcPDF está dividido en unidades por funcionalidad. Solo agregue a la cláusula *uses* las unidades cuyos tipos utilice.

Unidades del núcleo

Unidad	Contenido
sgcPDF_Document	TsgcPDFDocument, TsgcPDFPage, TsgcPDFDocumentInfo
sgcPDF_Editor	TsgcPDFEditor, TsgcPDFWatermarkOptions, sgcPDFMergeFiles, sgcPDFExtractPages, sgcPDFGetPageCount
sgcPDF_Fonts	TsgcPDFStandardFont (las 14 fuentes estándar), sgcPDFMeasureTextWidth
sgcPDF_Fonts_TrueType	TsgcTrueTypeFont: análisis, subconjunto, incrustación y shaping de TrueType y OpenType (CFF)
sgcPDF_Shaping	TsgcPDFShaper: shaping OpenType GSUB/GPOS, unión árabe, algoritmo bidireccional de Unicode
sgcPDF_Fonts_CFF, sgcPDF_Fonts_Type1	Programas de fuente CFF y Type 1 utilizados por los renderizadores
sgcPDF_Fonts_CJK	TsgcPDFCIDFont, TsgcPDFCJKFontManager (bajo nivel, solo Windows)
sgcPDF_ContentStream	TsgcPDFContentBuilder (operadores de dibujo de página), TsgcPDFContentParser
sgcPDF_Graphics	TsgcPDFExtGState, TsgcPDFShading, TsgcPDFPattern, TsgcPDFGraphicsResources
sgcPDF_Canvas	TsgcPDFCanvas: dibujo estilo TCanvas en páginas PDF
sgcPDF_HTML	TsgcPDFHTMLRenderer: HTML y CSS a PDF
sgcPDF_Images	TsgcPDFImage: carga de JPEG, PNG, BMP y TIFF
sgcPDF_JPEG, sgcPDF_JPX, sgcPDF_CCITT, sgcPDF_JBIG2	Decodificadores de imagen: JPEG, JPEG 2000, CCITT fax y JBIG2
sgcPDF_ColorSpaces	Evaluación de espacio de color y funciones compartida por los renderizadores y el extractor de imágenes
sgcPDF_ImageExtract	TsgcPDFImageExtractor, sgcPDFEncodePNG, sgcPDFEncodeBMP
sgcPDF_Tables	TsgcPDFTable, TsgcPDFTableRow, TsgcPDFTableCell
sgcPDF_Barcodes	TsgcPDFBarcodeGenerator, TsgcPDFBarcodeOptions: códigos de barras lineales, QR Code, DataMatrix, PDF417, GS1-128
sgcPDF_Barcode_Symbol	TsgcPDFBarcodeSymbol: diseño de código de barras independiente del framework con salida SVG, PNG y PDF

Unidad	Contenido
sgcPDF_Annotations	TsgcPDFAnnotationManager y las clases de anotación
sgcPDF_Bookmarks	TsgcPDFBookmarks, TsgcPDFBookmarkItem
sgcPDF_Layers	TsgcPDFLayerManager, TsgcPDFLayer, TsgcPDFLayerGroup
sgcPDF_Forms	TsgcPDFAcroForm, TsgcPDFFormField
sgcPDF_Attachments	TsgcPDFAttachments: archivos incrustados y adjuntos PDF/A-3
sgcPDF_Tagged	Árbol de estructura de PDF etiquetado
sgcPDF_Redaction	TsgcPDFRedactor
sgcPDF_Encryption	TsgcPDFEncryptionMethod, TsgcPDFEncryptionHandler, funciones auxiliares de hash y cifrado
sgcPDF_Signatures	TsgcPDFSignatureHandler, TsgcPDFSignatureVerifier
sgcPDF_PDFA	TsgcPDFALevel, TsgcPDFAValidator, TsgcPDFAConverter, TsgcPDFXMPMetadata
sgcPDF_Optimize	TsgcPDFOptimizer, TsgcPDFOptimizeOptions, sgcPDFOptimizeFile
sgcPDF_TextExtract	TsgcPDFTextExtractor, sgcPDFExtractPageText, sgcPDFExtractAllText, sgcPDFFindText, sgcPDFCreatePageTextExtractor
sgcPDF_Renderer	TsgcPDFRenderer: el intérprete de contenido independiente de la plataforma
sgcPDF_Renderer_Canvas	TsgcPDFRenderCanvas: el backend de dibujo abstracto
sgcPDF_Renderer_Fonts	Programas de fuente y contornos de glifo para los renderizadores
sgcPDF_Renderer_Raster	TsgcPDFRendererRaster, TsgcPDFRasterCanvas, sgcPDFRasterAddFontDirectory
sgcPDF_Parser	TsgcPDFParser: el lector de PDF de bajo nivel
sgcPDF_Writer	TsgcPDFWriter: el escritor de PDF de bajo nivel
sgcPDF_Filters	Filtros de stream (Flate, LZW, ASCII85, ASCIIHex, RunLength, CCITT, JBIG2, predictores) y sgcPDFMaxDecodedSize
sgcPDF_Types	Modelo de objetos de PDF (TsgcPDFDictionary, TsgcPDFArray, TsgcPDFStream...), TsgcPDFRect, TBytes

Unidades VCL (Windows)

Unidad	Contenido
sgcPDF_Renderer_GDI	TsgcPDFRendererGDI, TsgcPDFGDICanvas: renderizado en un contexto de dispositivo de Windows
sgcPDF_Export	TsgcPDFExporter, TsgcPDFExportOptions, sgcPDFPageToImage, sgcPDFToImages

Unidad	Contenido
sgcPDF_Viewer	TsgcPDFViewer
sgcPDF_ViewerVCL	TsgcPDFViewerControl, TsgcPDFThumbnailView, TsgcPDFOutlineView
sgcPDF_EMF	TsgcPDFEMFPlayer, sgcPDFEMFAddPage
sgcPDF_CanvasVCL	TsgcPDFVCLCanvasHelper
sgcPDF_Barcode_VCL	TsgcPDFBarcodeControl

Unidades FireMonkey (XE5 y posterior)

Unidad	Contenido
sgcPDF_Renderer_FMX	TsgcPDFRendererFMX, TsgcPDFFMXCanvas, sgcPDFBGRAToFMXBitmap
sgcPDF_ViewerFMX	TsgcPDFViewerFMX
sgcPDF_Barcode_FMX	TsgcPDFBarcodeFMX, TsgcPDFBarcodeTextSettingsFMX

La unidad opcional sgcPDF_Sign_sgcSign (firmas PAdES mediante sgcSign) no forma parte de los paquetes. Agréguela a su proyecto cuando la utilice.

Coordenadas y unidades

Todas las posiciones y tamaños están en puntos PDF (1/72 de pulgada). El origen es la esquina **inferior izquierda** de la página y el eje Y crece hacia arriba. Una página A4 mide 595 x 842 puntos y una página US Letter mide 612 x 792 puntos. Los rectángulos pasados como *Left, Bottom, Right, Top* utilizan el mismo espacio de coordenadas. Los colores se indican como componentes R, G, B entre 0 y 1. TsgcPDFCanvas es la excepción: utiliza un origen superior izquierdo y la unidad que usted elija, como un TCanvas de VCL.

Índices de página

Los índices de página tienen base cero: la primera página es 0 y la última es `PageCount - 1`.

Cadenas

Los métodos que dibujan con las fuentes estándar toman `AnsiString` (codificación WinAnsi). Los métodos que dibujan con fuentes TrueType, el texto extraído, el contenido de las anotaciones, los títulos de marcadores y los valores de formulario utilizan `WideString`, de modo que admiten cualquier carácter Unicode que contenga la fuente.

Inicio rápido

Los siguientes ejemplos son procedimientos completos tomados de programas de consola que fueron compilados y ejecutados. Agregue las unidades enumeradas en cada sección a su cláusula *uses*.

Crear un PDF

Unidades: sgcPDF_Fonts, sgcPDF_Document.

```
procedure CreateHelloPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.Info.Title := 'Hello sgcPDF';
    oPDF.Info.Author := 'eSeGeCe';
    // A4 portrait: 595 x 842 points, origin at the bottom left corner
    oPage := oPDF.AddPage(595, 842);
    oPage.DrawTextColor('Hello sgcPDF', 50, 780, sfHelveticaBold, 24, 0, 0.2, 0.6);
    oPage.DrawLine(50, 770, 545, 770, 1.5);
    oPage.DrawText('A PDF written in pure Pascal.', 50, 740, sfHelvetica, 12);
    oPage.DrawRectangle(40, 40, 515, 762, 1.0);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Abrir y editar un PDF

Unidades: sgcPDF_Fonts, sgcPDF_Editor. TsgcPDFEditor carga un documento, aplica cambios a cada página y lo guarda.

```
procedure StampExistingPDF(const AInput, AOutput: string);
var
  oEditor: TsgcPDFEditor;
  oWatermark: TsgcPDFWatermarkOptions;
begin
  oEditor := TsgcPDFEditor.Create;
  try
    oEditor.LoadFromFile(AInput);
    oWatermark.Text := 'CONFIDENTIAL';
    oWatermark.Font := sfHelveticaBold;
    oWatermark.FontSize := 48;
    oWatermark.Rotation := 45;
    oWatermark.Opacity := 0.3;
    oWatermark.R := 0.5;
    oWatermark.G := 0.5;
    oWatermark.B := 0.5;
    oEditor.AddWatermarkAllPages(oWatermark);
    oEditor.AddHeaderText('Internal report', sfHelvetica, 9);
    oEditor.AddPageNumbers(sfHelvetica, 10, 'Page %d of %d');
    oEditor.AddText(0, 'Reviewed', 450, 60, sfHelveticaOblique, 11);
    oEditor.SaveToFile(AOutput);
  finally
    oEditor.Free;
  end;
end;
```

Extraer texto

Unidades: sgcPDF_Document.

```
procedure PrintAllText(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    for i := 0 to oPDF.PageCount - 1 do
    begin
      WriteLn('--- page ', i + 1, ' ---');
      WriteLn(string(oPDF.ExtractTextW(i)));
    end;
    if oPDF.FindText(0, 'Hello') then
      WriteLn('Page 1 contains "Hello"');
  finally
    oPDF.Free;
  end;
end;
```

Renderizar una página a una imagen

Unidades: sgcPDF_Document, sgcPDF_Export.

```
procedure ExportPages(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oExporter: TsgcPDFExporter;
  oOptions: TsgcPDFExportOptions;
begin
  oPDF := TsgcPDFDocument.Create;
  oExporter := TsgcPDFExporter.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oOptions := sgcPDFDefaultExportOptions;
    oOptions.Format := efPNG;
    oOptions.DPI := 150;
    oExporter.ExportPage(oPDF, 0, 'out\page1.png', oOptions);
    oExporter.ExportThumbnail(oPDF, 0, 'out\thumb1.bmp', 200, 200);
  finally
    oExporter.Free;
    oPDF.Free;
  end;
  // one call helpers that open the file themselves
  sgcPDFPageToImage(AFileName, 'out\page1_96.png', 0, 96);
end;
```

Ver un PDF

Unidades: sgcPDF_Viewer. TsgcPDFViewer abre una ventana nativa. En un programa de consola, `RunMessageLoop` mantiene la ventana activa hasta que se cierra.

```
procedure ShowViewer(const AFileName: string);
var
  oViewer: TsgcPDFViewer;
begin
  oViewer := TsgcPDFViewer.Create;
  try
    oViewer.ViewMode := vmContinuous;
    // top level window; use CreateWindow(AParent, X, Y, W, H) to embed it
```

```
    oViewer.CreateWindow(100, 100, 800, 900, 'My PDF Viewer');
    oViewer.LoadFromFile(AFileName);
    oViewer.ZoomFit;
    oViewer.Show;
    oViewer.RunMessageLoop;
  finally
    oViewer.Free;
  end;
end;
```

Vea también

- Texto y fuentes
- Renderizado y exportación

TsgcPDFDocument

Unidad: sgcPDF_Document
La clase central de sgcPDF. Contiene las páginas de un documento, dibuja contenido nuevo, carga archivos existentes y guarda el resultado, opcionalmente cifrado o como PDF/A. También da acceso a los marcadores, las anotaciones, los campos de formulario y las capas del documento.

Creación y carga

Método	Descripción
Create / Create(AOwner)	Crea un documento vacío.
LoadFromFile(const AFileName: string)	Carga un archivo PDF. Llame primero a SetPassword para archivos cifrados.
LoadFromStream(AStream: TStream)	Carga un PDF desde un stream.
LoadFromBytes(const AData: TBytes)	Carga un PDF desde memoria.
SaveToFile(const AFileName: string)	Escribe el documento en un archivo.
SaveToStream(AStream: TStream)	Escribe el documento en un stream.

Páginas

Miembro	Descripción
AddPage: TsgcPDFPage	Agrega una página A4 (595 x 842 puntos).
AddPage(AWidth, AHeight: Double): TsgcPDFPage	Agrega una página del tamaño indicado en puntos.
InsertPage(AIndex: Integer; AWidth, AHeight: Double): TsgcPDFPage	Inserta una página en una posición.
DeletePage(AIndex: Integer)	Elimina una página.
GetPage(AIndex: Integer) / Pages[AIndex]	Devuelve una página (propiedad predeterminada).
PageCount: Integer	Número de páginas.
AppendDocument(ADoc: TsgcPDFDocument)	Agrega las páginas de otro documento.

Recursos

Método	Descripción
AddTrueTypeFont(const AFontName: WideString; ABold, AItalic: Boolean): Integer	Carga una fuente TrueType instalada y devuelve su índice de fuente para TsgcPDFPage.DrawTextTT.
AddImage(const AFileName: string): Integer	Carga un archivo JPEG, PNG o BMP y devuelve su índice de imagen.
AddImageFromStream(AStream: TStream): Integer	Igual que AddImage, desde un stream.

Texto

Método	Descripción
ExtractText(APageIndex: Integer): AnsiString	Texto de una página como AnsiString.
ExtractTextW(APageIndex: Integer): WideString	Texto de una página como Unicode.
FindText(APageIndex: Integer; const AText: WideString): Boolean	True cuando la página contiene el texto.
GetPageContentStream(APageIndex: Integer): TBytes	El stream de contenido decodificado de una página.

Seguridad y conformidad

Método	Descripción
SetEncryption(AMethod: TsgcPDFEncryptionMethod; const AUserPassword, AOwnerPassword: AnsiString)	Cifra el documento al guardarlo.
SetPassword(const APassword: AnsiString)	Contraseña utilizada para abrir un archivo cifrado. Llámelo antes de LoadFromFile.
SetPDFALevel(ALevel: TsgcPDFALevel)	Guarda el documento como PDF/A-1b, PDF/A-2b o PDF/A-3b.

Propiedades

Propiedad	Tipo	Descripción
Info	TsgcPDFDocumentInfo	Title, Author, Subject, Keywords, Creator y Producer.
Bookmarks	TsgcPDFBookmarks	Esquema del documento. Vea Marcadores.
Annotations	TsgcPDFAnnotationManager	Anotaciones de todas las páginas. Vea Anotaciones.
AcroForm	TsgcPDFAcroForm	Campos de formulario interactivo. Vea Formularios.
Layers	TsgcPDFLayerManager	Grupos de contenido opcional. Vea Capas.

Propiedad	Tipo	Descripción
GraphicsResources	TsgcPDFGraphicsResources	Estados de transparencia, degradados y patrones.
IsLoaded	Boolean	True después de cargar un documento.
Parser	TsgcPDFParser	El analizador de un documento cargado, utilizado por el extractor de texto, el validador y el renderizador.
PDFALevel	TsgcPDFALevel	El nivel de PDF/A establecido con SetPDFALevel.
Version (publicada)	AnsiString	Versión de PDF escrita en el encabezado del archivo.

TsgcPDFPage

Devuelto por AddPage y GetPage. Sus métodos de dibujo agregan operadores al contenido de la página.

Método	Descripción
DrawText(const AText: AnsiString; X, Y: Double; AFont: TsgcPDFStandardFont; AFontSize: Double)	Texto con una fuente estándar, comenzando en el punto de línea base X, Y.
DrawTextColor(..., R, G, B: Double)	Igual, con un color de texto.
DrawTextTT(const AText: WideString; X, Y: Double; AFontIndex: Integer; AFontSize: Double)	Texto Unicode con una fuente TrueType agregada con AddTrueTypeFont.
DrawTextTTColor(..., R, G, B: Double)	Igual, con un color de texto.
DrawLine(X1, Y1, X2, Y2: Double; ALineWidth: Double = 1.0)	Línea. DrawLineColor agrega un color.
DrawRectangle(X, Y, W, H: Double; ALineWidth: Double = 1.0)	Contorno de rectángulo. DrawRectangleFill lo rellena con un color.
DrawCircle(CX, CY, Radius: Double; ALineWidth: Double = 1.0)	Contorno de círculo. DrawCircleFill lo rellena con un color.
DrawImageFromIndex(AImageIndex: Integer; X, Y, W, H: Double)	Dibuja una imagen agregada con AddImage.
DrawImageFile(const AFileName: string; X, Y, W, H: Double)	Carga y dibuja una imagen en una sola llamada.
SetOpacity(AOpacity: Double)	Opacidad (0 a 1) para las siguientes operaciones.
DrawRectangleTransparent(X, Y, W, H: Double; R, G, B, AOpacity: Double)	Rectángulo relleno semitransparente.
DrawLinearGradient(X, Y, W, H: Double; R0, G0, B0, R1, G1, B1: Double)	Rectángulo relleno con un degradado lineal.

Propiedades de página: **Width**, **Height**, **MediaBox**, **PageIndex**, **PageDict** y **Content**, el TsgcPDFContentBuilder que da acceso a todos los operadores de dibujo (vea Gráficos).

Eventos

TsgcPDFDocument no tiene eventos.

Ejemplo

```
procedure CreateHelloPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.Info.Title := 'Hello sgcPDF';
    oPDF.Info.Author := 'eSeGeCe';
    // A4 portrait: 595 x 842 points, origin at the bottom left corner
    oPage := oPDF.AddPage(595, 842);
    oPage.DrawTextColor('Hello sgcPDF', 50, 780, sfHelveticaBold, 24, 0, 0.2, 0.6);
    oPage.DrawLine(50, 770, 545, 770, 1.5);
    oPage.DrawText('A PDF written in pure Pascal.', 50, 740, sfHelvetica, 12);
    oPage.DrawRectangle(40, 40, 515, 762, 1.0);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- Texto y fuentes
- Combinar, dividir y extraer páginas

TsgcPDFEditor

Unidad: sgcPDF_Editor
Edición de alto nivel de un PDF existente. El editor envuelve un TsgcPDFDocument (propiedad **Document**) y añade operaciones de una sola llamada: marcas de agua, encabezados, pies de página, números de página, combinación, extracción de páginas, relleno de formularios y aplanado.

Constructores

Constructor	Descripción
Create / Create(AOwner: TComponent)	Crea el editor con su propio documento.
Create(ADocument: TsgcPDFDocument)	Edita una instancia de TsgcPDFDocument existente.

Métodos

Método	Descripción
LoadFromFile / SaveToFile	Carga y guarda el documento.
AddBlankPage / AddBlankPage(AWidth, AHeight: Double)	Agrega una página vacía.
DeletePage(AIndex: Integer)	Elimina una página.
RotatePage(AIndex: Integer; ADegrees: Integer)	Rota una página. ADegrees debe ser 90, 180 o 270.
AddText / AddTextColor(APageIndex, AText, X, Y, AFont, AFontSize [, R, G, B])	Dibuja texto en una página.
AddTransparentText(APageIndex, AText, X, Y, AFont, AFontSize, AOpacity)	Dibuja texto semitransparente.
AddLine / AddRectangle / AddRectangleFill	Dibuja formas en una página.
AddImageToPage(APageIndex: Integer; const AImageFile: string; X, Y, W, H: Double)	Dibuja un archivo de imagen en una página.
AddWatermark(APageIndex; const AOptions: TsgcPDFWatermarkOptions)	Texto rotado y semitransparente en una página.
AddWatermarkAllPages(const AOptions: TsgcPDFWatermarkOptions)	Igual, en todas las páginas.
AddHeaderText / AddFooterText(const AText: AnsiString; AFont; AFontSize)	Texto en la parte superior o inferior de todas las páginas.
AddPageNumbers(AFont; AFontSize: Double; const AFormat: AnsiString)	Números de página en todas las páginas. AFormat admite dos %d: página actual y número total de páginas.
AddTextAnnotation / AddLinkAnnotation / AddHighlightAnnotation	Agrega anotaciones.

Método	Descripción
AddBookmark(const ATitle: WideString; APageIndex: Integer)	Agrega un marcador de nivel superior. AddBookmarkChild agrega uno anidado.
MergeFrom(const AFileName: string)	Agrega todas las páginas de otro PDF.
ExtractPages(AFromIndex, AToIndex: Integer; const AOutputFileName: string)	Guarda un rango de páginas en un archivo nuevo.
ExtractText(APageIndex): AnsiString / ExtractTextUnicode(APageIndex): WideString / ExtractAllText	Extracción de texto.
SearchText(APageIndex: Integer; const AText: WideString): Boolean	True cuando la página contiene el texto.
FillFormField(const AFieldName: AnsiString; const AValue: WideString)	Establece el valor de un campo de formulario.
FlattenForm	Convierte los campos de formulario en contenido estático de página.
SetEncryption(AMethod; const AUserPassword, AOwnerPassword: AnsiString)	Cifra la salida.
ConvertToPDFA(ALevel: TsgcPDFALevel)	Guarda la salida como PDF/A.
SignDocument(const AOutputFile: string; const ACertSubjectName, AReason, ALocation: WideString)	Guarda y firma con un certificado del almacén de Windows, localizado por nombre de asunto.

TsgcPDFWatermarkOptions

Registro con los campos Text, Font, FontSize, Rotation (grados), Opacity (0 a 1) y R, G, B.

Funciones auxiliares

Función	Descripción
sgcPDFMergeFiles(const AInputFiles: array of string; const AOutputFile: string)	Combina varios archivos.
sgcPDFExtractPages(const AInputFile, AOutputFile: string; AFromPage, AToPage: Integer)	Extrae un rango de páginas con base cero.
sgcPDFGetPageCount(const AFileName: string): Integer	Número de páginas de un archivo.

Eventos

TsgcPDFEditor no tiene eventos.

Ejemplo

```
procedure StampExistingPDF(const AInput, AOutput: string);
var
  oEditor: TsgcPDFEditor;
  oWatermark: TsgcPDFWatermarkOptions;
begin
  oEditor := TsgcPDFEditor.Create;
  try
    oEditor.LoadFromFile(AInput);
    oWatermark.Text := 'CONFIDENTIAL';
    oWatermark.Font := sfHelveticaBold;
    oWatermark.FontSize := 48;
    oWatermark.Rotation := 45;
    oWatermark.Opacity := 0.3;
    oWatermark.R := 0.5;
    oWatermark.G := 0.5;
    oWatermark.B := 0.5;
    oEditor.AddWatermarkAllPages(oWatermark);
    oEditor.AddHeaderText('Internal report', sfHelvetica, 9);
    oEditor.AddPageNumbers(sfHelvetica, 10, 'Page %d of %d');
    oEditor.AddText(0, 'Reviewed', 450, 60, sfHelveticaOblique, 11);
    oEditor.SaveToFile(AOutput);
  finally
    oEditor.Free;
  end;
end;
```

Vea también

- Combinar, dividir y extraer páginas
- Formularios

TsgcPDFViewer

Unidad: sgcPDF_Viewer
Un visor de PDF implementado como una ventana nativa de Windows. No depende de la VCL, por lo que funciona en aplicaciones VCL (incrustado en cualquier control con ventana mediante su Handle) y en programas de consola o no visuales (como ventana de nivel superior). Las páginas se dibujan con el renderizador GDI.

Ventana

Método	Descripción
CreateWindow(AParent: HWND; X, Y, W, H: Integer)	Crea el visor como una ventana hija, por ejemplo de un TPanel.Handle.
CreateWindow(X, Y, W, H: Integer; const ATitle: AnsiString)	Crea una ventana de nivel superior.
Show / Hide / Invalidate / DestroyWindow	Gestión de la ventana.
RunMessageLoop	Ejecuta un bucle de mensajes hasta que se cierra la ventana. Utilícelo solo en programas sin su propio bucle de mensajes, como los programas de consola.

Documento y navegación

Método	Descripción
LoadFromFile(const AFileName: string)	Abre un archivo PDF.
LoadDocument(ADocument: TsgcPDFDocument; AOwns: Boolean = False)	Muestra un documento ya en memoria.
CloseDocument	Cierra el documento.
FirstPage / LastPage / NextPage / PrevPage / GotoPage(AIndex)	Navegación.
ZoomIn / ZoomOut / ZoomFit / ZoomFitPage / SetZoom(APercent: Double)	Zoom.
FindText(const AText: WideString): Boolean / FindNext / ClearSearch	Búsqueda de texto.
Print / PrintPage(APageIndex: Integer)	Impresión.
GetPageText(APageIndex: Integer): WideString	Texto de una página.
ExportPageToBitmap(APageIndex: Integer; const AFileName: string; AWidth, AHeight: Integer)	Renderiza una página a un archivo BMP.

La ventana también responde al teclado: Izquierda y Av Pág van a la página anterior, Derecha y Re Pág a la siguiente, Inicio y Fin a la primera y última página, las teclas más y menos hacen zoom, y Arriba y Abajo desplazan la vista.

Propiedades

Propiedad	Tipo	Descripción
ViewMode (publicada)	TsgcPDFViewer-Mode	vmSinglePage (predeterminado) o vmContinuous.
BackColor (publicada)	COLORREF	Color de fondo alrededor de las páginas. Predeterminado $00C8C8C8.
ShowAnnotations (publicada)	Boolean	Dibuja las anotaciones. Predeterminado True.
Handle	HWND	Identificador (handle) de la ventana.
CurrentPage	Integer	Índice con base cero de la página actual.
PageCount	Integer	Número de páginas.
Zoom	Double	Zoom actual.
Document	TsgcPDFDocu-ment	El documento que se muestra.

Eventos

TsgcPDFViewer no tiene eventos.

Ejemplo

```
procedure ShowViewer(const AFileName: string);
var
  oViewer: TsgcPDFViewer;
begin
  oViewer := TsgcPDFViewer.Create;
  try
    oViewer.ViewMode := vmContinuous;
    // top level window; use CreateWindow(AParent, X, Y, W, H) to embed it
    oViewer.CreateWindow(100, 100, 800, 900, 'My PDF Viewer');
    oViewer.LoadFromFile(AFileName);
    oViewer.ZoomFit;
    oViewer.Show;
    oViewer.RunMessageLoop;
  finally
    oViewer.Free;
  end;
end;
```

Vea también

- Renderizado y exportación

TsgcPDFTextExtractor

Unidad: sgcPDF_TextExtract
Extrae el texto de una página, conservando la posición, el nombre de fuente, el tamaño de fuente y el ancho de cada fragmento de texto. Decodifica codificaciones estándar, Differences de fuente, mapas ToUnicode y fuentes CID, de modo que el texto escrito con fuentes TrueType y CJK se devuelve como Unicode.

Constructores

Constructor	Descripción
Create(AParser: TsgcPDFParser; APageDict: TsgcPDFDictionary)	Prepara la extracción de una página. Pase TsgcPDFDocument.Parser y TsgcPDFPage.PageDict de un documento cargado.
Create(AOwner: TComponent)	Constructor de componente, utilizado por el IDE. Utilice el constructor anterior para extraer texto en código.

Métodos

Método	Descripción
Extract	Analiza el contenido de la página. Llámelo antes que los demás métodos.
GetText: WideString	El texto de la página en orden de lectura, una línea por línea de texto.
GetTextWithPositions: WideString	El texto con las coordenadas de cada fragmento.
GetFragmentCount: Integer	Número de fragmentos.
GetFragment(AIndex: Integer): TsgcPDFTextFragment	Un fragmento: Text, X, Y, FontSize, FontName y Width.
FindText(const ASearchText: WideString; ACaseSensitive: Boolean = False): Boolean	Busca en el texto de la página.

Funciones auxiliares

Función	Descripción
sgcPDFExtractPageText(AParser: TsgcPDFParser; APageIndex: Integer): WideString	Texto de una página.
sgcPDFExtractAllText(AParser: TsgcPDFParser): WideString	Texto de todas las páginas.
sgcPDFFindText(AParser: TsgcPDFParser; APageIndex: Integer; const AText: WideString): Boolean	Busca en una página.

Un TsgcPDFParser utilizado directamente debe cargarse y luego analizarse: llame a `LoadFromFile` y después a `Parse`.

Eventos

TsgcPDFTextExtractor no tiene eventos.

Ejemplo

```
procedure PrintTextPositions(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oExtractor: TsgcPDFTextExtractor;
  oFragment: TsgcPDFTextFragment;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oExtractor := TsgcPDFTextExtractor.Create(oPDF.Parser,
      oPDF.Pages[0].PageDict);
    try
      oExtractor.Extract;
      for i := 0 to oExtractor.GetFragmentCount - 1 do
      begin
        oFragment := oExtractor.GetFragment(i);
        WriteLn(Format('%.1f, %.1f  %s  (%s %.1f pt)', [oFragment.X, oFragment.Y,
          string(oFragment.Text), string(oFragment.FontName), oFragment.FontSize]));
      end;
    finally
      oExtractor.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- Extracción de texto

TsgcPDFExporter

Unidad: sgcPDF_Export
Renderiza las páginas de un TsgcPDFDocument a imágenes BMP, PNG o JPEG utilizando el renderizador GDI (TsgcPDFRendererGDI).

Métodos

Método	Descripción
ExportPage(ADocument: TsgcPDFDocument; APageIndex: Integer; const AFileName: string)	Renderiza una página con la propiedad Options.
ExportPage(ADocument; APageIndex; const AFileName: string; const AOptions: TsgcPDFExportOptions)	Renderiza una página con opciones explícitas.
ExportPageToStream(ADocument; APageIndex; AStream: TStream; const AOptions)	Renderiza una página a un stream.
ExportAllPages(ADocument; const AOutputDir, AFilePrefix: string [; const AOptions])	Renderiza todas las páginas a una carpeta.
ExportPages(ADocument; AFromPage, AToPage: Integer; const AOutputDir, AFilePrefix: string; const AOptions)	Renderiza un rango de páginas.
ExportThumbnail(ADocument; APageIndex; const AFileName: string; AMaxWidth, AMaxHeight: Integer)	Renderiza una página escalada para ajustarse al cuadro indicado.
ExportAllThumbnails(ADocument; const AOutputDir, AFilePrefix: string; AMaxWidth, AMaxHeight: Integer)	Miniaturas de todas las páginas.

TsgcPDFExportOptions

Campo	Predeterminado	Descripción
Format	efPNG	efBMP, efPNG o efJPEG.
DPI	150	Resolución. 72 DPI renderiza un píxel por punto.
JPEGQuality	85	Calidad JPEG, de 1 a 100.
BackgroundColor	$00FFFFFF	Fondo de página (COLORREF).
AntiAlias	False	Suaviza la salida.

Obtenga los valores predeterminados con `sgcPDFDefaultExportOptions`.

Funciones auxiliares

Función	Descripción
sgcPDFPageToImage(const APDFFile, AImageFile: string; APageIndex: Integer; ADPI: Integer = 150)	Abre un archivo y renderiza una página.
sgcPDFToImages(const APDFFile, AOutputDir: string; AFormat: TsgcPDFExportFormat = efPNG; ADPI: Integer = 150)	Abre un archivo y renderiza todas las páginas.

Eventos

TsgcPDFExporter no tiene eventos.

Ejemplo

```
procedure ExportPages(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oExporter: TsgcPDFExporter;
  oOptions: TsgcPDFExportOptions;
begin
  oPDF := TsgcPDFDocument.Create;
  oExporter := TsgcPDFExporter.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oOptions := sgcPDFDefaultExportOptions;
    oOptions.Format := efPNG;
    oOptions.DPI := 150;
    oExporter.ExportPage(oPDF, 0, 'out\page1.png', oOptions);
    oExporter.ExportThumbnail(oPDF, 0, 'out\thumb1.bmp', 200, 200);
  finally
    oExporter.Free;
    oPDF.Free;
  end;
  // one call helpers that open the file themselves
  sgcPDFPageToImage(AFileName, 'out\page1_96.png', 0, 96);
end;
```

Vea también

- Renderizado y exportación

TsgcPDFOptimizer

Unidad: sgcPDF_Optimize
Reescribe un archivo PDF para reducir su tamaño: comprime los streams sin comprimir, elimina objetos duplicados y no referenciados, y puede reducir la resolución de las imágenes y eliminar contenido opcional como metadatos, marcadores, anotaciones, JavaScript y miniaturas.

Métodos

Método	Descripción
Optimize(const AInputFile, AOutputFile: string)	Optimiza un archivo.
OptimizeStream(AInputStream, AOutputStream: TStream)	Optimiza un stream.
GetSizeReduction: Int64	Bytes ahorrados por la última optimización.
GetLog: TStringList	Informe de la última optimización. Es propiedad del optimizador, no lo libere.

TsgcPDFOptimizeOptions

Campo	Predeterminado	Descripción
CompressStreams	True	Flate comprime los streams sin comprimir.
RemoveDuplicateObjects	True	Combina objetos idénticos.
RemoveUnusedObjects	True	Descarta los objetos no alcanzables desde el documento.
DownsampleImages	False	Reduce las imágenes por encima de MaxImageDPI.
MaxImageDPI	150	Resolución objetivo para la reducción de muestreo.
JPEGQuality	75	Calidad de las imágenes recomprimidas.
RemoveMetadata	False	Elimina los metadatos XMP.
RemoveBookmarks	False	Elimina el esquema.
RemoveAnnotations	False	Elimina las anotaciones.
RemoveJavaScript	False	Elimina las acciones de JavaScript.
FlattenForms	False	Aplana los campos de formulario.
RemoveThumbnails	False	Elimina las miniaturas de página incrustadas.

Obtenga los valores predeterminados con `sgcPDFDefaultOptimizeOptions`, u optimice un archivo con los valores predeterminados en una sola llamada con `sgcPDFOptimizeFile(const AInputFile, AOutputFile: string)`.

Eventos

TsgcPDFOptimizer no tiene eventos.

Ejemplo

```
procedure OptimizePDF(const AInput, AOutput: string);
var
  oOptimizer: TsgcPDFOptimizer;
  oOptions: TsgcPDFOptimizeOptions;
begin
  oOptimizer := TsgcPDFOptimizer.Create;
  try
    oOptions := sgcPDFDefaultOptimizeOptions;
    oOptions.DownsampleImages := True;
    oOptions.MaxImageDPI := 150;
    oOptions.RemoveMetadata := True;
    oOptimizer.Options := oOptions;
    oOptimizer.Optimize(AInput, AOutput);
    WriteLn('Saved ', oOptimizer.GetSizeReduction, ' bytes');
    WriteLn(oOptimizer.GetLog.Text);
  finally
    oOptimizer.Free;
  end;
end;
```

Vea también

- Optimización de archivos PDF

TsgcPDFSignatureHandler

Unidad: sgcPDF_Signatures
Firma documentos PDF con un certificado X.509 y su clave privada. La firma es una firma PKCS#7 (CMS) separada (*/SubFilter /adbe.pkcs7.detached*) creada mediante CryptoAPI de Windows, y se agrega como una actualización incremental para que los bytes originales permanezcan sin cambios.

Certificado

Método	Descripción
LoadCertificateFromPFX(const AFileName: string; const APassword: AnsiString)	Carga un certificado y una clave privada desde un archivo PFX (PKCS#12).
LoadCertificateFromStore(const ASubjectName: WideString)	Busca un certificado por nombre de asunto en el almacén MY del usuario actual.
LoadCertificateFromThumbprint(const AThumbprint: AnsiString)	Busca un certificado por huella digital SHA-1.
ListCertificates: TStringList	Certificados del almacén. El llamador libera la lista.
GetCertificateInfo: TsgcPDFCertificateInfo	SubjectName, IssuerName, SerialNumber, NotBefore, NotAfter y Thumbprint del certificado cargado.

Firma

Método	Descripción
SetField(const AName: AnsiString; APageIndex: Integer; ALeft, ABottom, ARight, ATop: Double)	Campo de firma visible en una página.
SetInvisible	Firma invisible.
SignDocument(const AInputFile, AOutputFile: string)	Firma un archivo.
SignDocumentStream(AInputStream, AOutputStream: TStream)	Firma un stream.

Propiedades

Propiedad	Tipo	Descripción
SignerName (publicada)	WideString	Nombre del firmante escrito en la firma. Igual que Name.
Reason (publicada)	WideString	Motivo de la firma.
Location (publicada)	WideString	Lugar de la firma.

Propiedad	Tipo	Descripción
ContactInfo (publicada)	AnsiString	Información de contacto.
HashAlgorithm (publicada)	TsgcPDFSignatureHashAlgorithm	shaSHA1, shaSHA256 (predeterminado), shaSHA384 o shaSHA512.
Appearance	TsgcPDFSignatureAppearance	Apariencia de firma visible: ShowName, ShowDate, ShowReason, ShowLocation, Text, FontName, FontSize, ImageData.
CertInfo	TsgcPDFCertificateInfo	Información del certificado cargado.

Eventos

TsgcPDFSignatureHandler no tiene eventos.

Ejemplo

```
procedure SignPDF(const AInput, AOutput: string);
var
  oSigner: TsgcPDFSignatureHandler;
begin
  oSigner := TsgcPDFSignatureHandler.Create;
  try
    oSigner.LoadCertificateFromPFX('signer.pfx', 'test123');
    WriteLn('Signing as ', string(oSigner.GetCertificateInfo.SubjectName));
    oSigner.Reason := 'Document approval';
    oSigner.Location := 'Barcelona';
    oSigner.HashAlgorithm := shaSHA256;
    // visible signature field on page 1: Left, Bottom, Right, Top
    oSigner.SetField('Signature1', 0, 50, 50, 250, 100);
    oSigner.SignDocument(AInput, AOutput);
  finally
    oSigner.Free;
  end;
end;
```

Vea también

- TsgcPDFSignatureVerifier
- Firmas digitales

TsgcPDFSignatureVerifier

Unidad: sgcPDF_Signatures
Verifica las firmas digitales de un documento PDF: comprueba que cada firma sea criptográficamente válida, si los rangos de bytes firmados aún cubren todo el archivo, y si el certificado del firmante encadena con una raíz de confianza en el almacén de certificados de Windows.

Métodos

Método	Descripción
LoadDocument(const AFileName: string)	Carga un PDF firmado.
GetSignatureCount: Integer	Número de firmas.
VerifySignature(AIndex: Integer): Boolean	True cuando la firma coincide con los bytes firmados.
IsDocumentModified(AIndex: Integer): Boolean	True cuando el archivo se modificó después de esta firma.
IsSignatureTrusted(AIndex: Integer): Boolean	True cuando la cadena de certificados es de confianza para Windows.
GetSignatureInfo(AIndex: Integer): TsgcPDFCertificateInfo	Información del certificado del firmante.

Un certificado de prueba autofirmado se verifica como válido pero no de confianza.

Eventos

TsgcPDFSignatureVerifier no tiene eventos.

Ejemplo

```
procedure VerifyPDF(const AFileName: string);
var
  oVerifier: TsgcPDFSignatureVerifier;
  oCert: TsgcPDFCertificateInfo;
  i: Integer;
begin
  oVerifier := TsgcPDFSignatureVerifier.Create;
  try
    oVerifier.LoadDocument(AFileName);
    for i := 0 to oVerifier.GetSignatureCount - 1 do
    begin
      oCert := oVerifier.GetSignatureInfo(i);
      WriteLn('Signer:   ', string(oCert.SubjectName));
      WriteLn('Valid:    ', oVerifier.VerifySignature(i));
      WriteLn('Modified: ', oVerifier.IsDocumentModified(i));
      WriteLn('Trusted:  ', oVerifier.IsSignatureTrusted(i));
    end;
  finally
    oVerifier.Free;
  end;
end;
```

Vea también

- TsgcPDFSignatureHandler
- Firmas digitales

Texto y fuentes

sgcPDF dibuja texto con dos tipos de fuentes: las 14 fuentes estándar de PDF, que proporciona todo lector de PDF, y las fuentes TrueType, que se incrustan en el archivo.

Fuentes estándar

Las fuentes estándar se seleccionan con **TsgcPDFStandardFont** (unidad sgcPDF_Fonts): sfCourier, sfCourierBold, sfCourierOblique, sfCourierBoldOblique, sfHelvetica, sfHelveticaBold, sfHelveticaOblique, sfHelveticaBoldOblique, sfTimesRoman, sfTimesBold, sfTimesItalic, sfTimesBoldItalic, sfSymbol y sfZapfDingbats. No se incrustan, por lo que mantienen los archivos pequeños, pero solo cubren el conjunto de caracteres WinAnsi. Utilícelas con `TsgcPDFPage.DrawText` y `DrawTextColor`.
`sgcPDFMeasureTextWidth(AFont, AText, AFontSize)` devuelve el ancho de una cadena en puntos, que es la forma de centrar o alinear texto a la derecha.

TrueType y Unicode

`TsgcPDFDocument.AddTrueTypeFont(AFontName, ABold, AItalic)` carga una fuente de Windows instalada por su nombre de familia y devuelve un índice de fuente. El texto dibujado con `DrawTextTT` o `DrawTextTTColor` es un WideString, de modo que se puede utilizar cualquier carácter contenido en la fuente. Al guardar el documento, solo se incrustan los glifos realmente utilizados (subconjunto de fuente) junto con un mapa ToUnicode, de modo que el texto se puede buscar, copiar y extraer.

```
procedure CreateFontsPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vArial, vTimes: Integer;
  vWidth: Double;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // TrueType fonts are loaded from the installed system fonts, subset and embedded
    vArial := oPDF.AddTrueTypeFont('Arial', False, False);
    vTimes := oPDF.AddTrueTypeFont('Times New Roman', True, False);
    oPage := oPDF.AddPage(595, 842);
    // the 14 standard fonts need no embedding
    oPage.DrawText('Standard Helvetica', 50, 780, sfHelvetica, 14);
    oPage.DrawText('Standard Courier', 50, 760, sfCourier, 14);
    // TrueType text is Unicode (WideString)
    oPage.DrawTextTT('Arial TrueType', 50, 720, vArial, 14);
    oPage.DrawTextTTColor('Times New Roman Bold', 50, 700, vTimes, 14, 0.6, 0, 0);
    oPage.DrawTextTT('Cyrillic: ' + WideString(#$041F#$0440#$0438#$0432#$0435#$0442),
      50, 670, vArial, 14);
    oPage.DrawTextTT('Greek: ' + WideString(#$0391#$03BB#$03C6#$03B1), 50, 650,
      vArial, 14);
    // measure a standard font string to right align it at x = 545
    vWidth := sgcPDFMeasureTextWidth(sfHelvetica, 'Right aligned', 12);
    oPage.DrawText('Right aligned', 545 - vWidth, 620, sfHelvetica, 12);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Texto CJK

El texto en chino, japonés y coreano funciona de la misma manera: agregue una fuente TrueType que contenga los caracteres y dibuje con DrawTextTT. La fuente se incrusta como una fuente CID con codificación Identity-H. La

enumeración **TsgcPDFCJKFont** (unidad sgcPDF_Fonts_CJK) enumera las fuentes CJK comunes de Windows y `sgcPDFCJKFontNameFromEnum` devuelve sus nombres de familia:

Valor	Fuente	Escritura
cjkMSGothic, cjkMSMincho, cjkMeiryo	MS Gothic, MS Mincho, Meiryo	Japonés
cjkSimSun, cjkSimHei, cjkMicrosoftYaHei	SimSun, SimHei, Microsoft YaHei	Chino simplificado
cjkMingLiU, cjkMicrosoftJhengHei	MingLiU, Microsoft JhengHei	Chino tradicional
cjkBatang, cjkGulim, cjkMalgunGothic	Batang, Gulim, Malgun Gothic	Coreano

```
procedure CreateCJKPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vKorean: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // any installed TrueType font that covers the script works the same way
    vKorean := oPDF.AddTrueTypeFont(sgcPDFCJKFontNameFromEnum(cjkMalgunGothic),
      False, False);
    oPage := oPDF.AddPage(595, 842);
    // "Annyeonghaseyo" in Hangul
    oPage.DrawTextTT(WideString(#$C548#$B155#$D558#$C138#$C694), 50, 780,
      vKorean, 20);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

La fuente debe estar instalada en el equipo que crea el PDF. El lector del PDF no la necesita, porque los glifos están incrustados.

Vea también

- TsgcPDFDocument
- Extracción de texto
- PDF/A

Imágenes

sgcPDF incrusta imágenes JPEG, PNG y BMP. Los datos JPEG se almacenan tal cual (DCTDecode) sin recompresión. Los píxeles de PNG y BMP se comprimen con Flate.
Agregue una imagen una vez con `TsgcPDFDocument.AddImage` (o `AddImageFromStream`) y dibújela tantas veces como sea necesario con `TsgcPDFPage.DrawImageFromIndex`. El archivo almacena la imagen una sola vez. `DrawImageFile` carga y dibuja un archivo en una sola llamada. X e Y son la esquina inferior izquierda de la imagen y W y H su tamaño en la página, en puntos.

```
procedure CreateImagesPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vLogo: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // JPEG is embedded as is (DCTDecode), PNG and BMP are recompressed
    vLogo := oPDF.AddImage('img.png');
    oPage := oPDF.AddPage(595, 842);
    // the image is added once and can be drawn many times
    oPage.DrawImageFromIndex(vLogo, 50, 600, 200, 100);
    oPage.DrawImageFromIndex(vLogo, 300, 600, 100, 50);
    // or load and draw a file in one call
    oPage.DrawImageFile('img.jpg', 50, 400, 300, 150);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

La clase **TsgcPDFImage** (unidad sgcPDF_Images) carga e inspecciona una imagen sin un documento: Width, Height, BitsPerComponent, ColorSpace, HasAlpha. Otros formatos, como TIFF, generan una excepción.

Vea también

- TsgcPDFDocument
- Optimización de archivos PDF

Gráficos, transparencia y degradados

TsgcPDFPage tiene funciones auxiliares para formas comunes: DrawLine, DrawRectangle, DrawRectangleFill, DrawCircle, DrawCircleFill y sus variantes con color. Para la transparencia utilice `SetOpacity` (afecta a las operaciones que le siguen) o `DrawRectangleTransparent`. `DrawLinearGradient` rellena un rectángulo con un degradado de color, dibujado como bandas de color finas desde el primer color hasta el segundo.

El generador de contenido

`TsgcPDFPage.Content` devuelve el **TsgcPDFContentBuilder** de la página (unidad sgcPDF_ContentStream). Expone los operadores de dibujo de PDF uno por uno, y todo lo que escribe va al mismo contenido de página que las funciones auxiliares:

- Estado gráfico: SaveState, RestoreState, SetMatrix, SetLineWidth, SetLineCap, SetLineJoin, SetMiterLimit, SetDashPattern.
- Colores: SetStrokeColorRGB, SetFillColorRGB, SetStrokeColorCMYK, SetFillColorCMYK, SetStrokeGray, SetFillGray.
- Trayectos: MoveTo, LineTo, CurveTo, CurveToV, CurveToY, Rectangle, ClosePath.
- Pintura: Stroke, CloseAndStroke, Fill, FillEvenOdd, FillAndStroke, EndPath, ClipNonZero, ClipEvenOdd.
- Texto: BeginText, EndText, SetFont, SetCharSpacing, SetWordSpacing, SetTextRise, SetTextRenderMode, SetTextScale, MoveTextPos, SetTextMatrix, ShowText.
- Otros: SetGraphicsState, DrawXObject, BeginMarkedContent, EndMarkedContent y AddOperator para escribir cualquier operador.

Empareje siempre SaveState con RestoreState para que los cambios no se filtren al dibujo posterior.

```
procedure CreateGraphicsPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  oContent: TsgcPDFContentBuilder;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    // shapes, gradients and transparency through the page helpers
    oPage.DrawLinearGradient(50, 680, 495, 100, 0.2, 0.4, 0.8, 0.9, 0.9, 1.0);
    oPage.DrawRectangleFill(70, 520, 200, 120, 1, 1, 0);
    oPage.DrawRectangleTransparent(150, 550, 200, 120, 1, 0, 0, 0.5);
    oPage.SetOpacity(0.6);
    oPage.DrawCircleFill(450, 580, 60, 0, 0.5, 0);
    oPage.SetOpacity(1.0);
    // low level path operators through the content builder
    oContent := oPage.Content;
    oContent.SaveState;
    oContent.SetStrokeColorCMYK(0, 1, 1, 0);
    oContent.SetLineWidth(3);
    oContent.SetDashPattern([6, 3], 0);
    oContent.MoveTo(50, 400);
    oContent.CurveTo(150, 500, 250, 300, 350, 400);
    oContent.Stroke;
    oContent.RestoreState;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFDocument
- Capas

Tablas

TsgcPDFTable (unidad sgcPDF_Tables) organiza filas y columnas, ajusta el texto dentro de las celdas y dibuja bordes y fondos.

- Créelo con el número de columnas y establezca los anchos con SetColumnWidths, SetColumnWidth o SetColumnWidthPercent.
- SetHeaders agrega una fila de encabezado. AddRowData agrega una fila de datos a partir de un array de cadenas. AddRow y AddHeaderRow devuelven un TsgcPDFTableRow para un control total sobre sus celdas (TsgcPDFTableCell con Style, ColSpan y RowSpan).
- Estilo: SetDefaultFont, SetHeaderFont, SetHeaderBackground, SetAlternateRowColors, SetBorder, SetPadding, o los registros DefaultStyle y HeaderStyle.

Dibujo

Método	Descripción
DrawOnPage(APage: TObject; AX, AY: Double)	Dibuja toda la tabla en un TsgcPDFPage. AX, AY es la esquina superior izquierda.
DrawOnPageWithOverflow(APage: TObject; AX, AY, AMaxHeight: Double): Integer	Dibuja las filas de encabezado y las filas de datos que caben en AMaxHeight, y devuelve el índice de la primera fila que no cupo.
DrawMultiPage(ADocument: TObject; AX, AY, AMaxHeight: Double; APageWidth, APageHeight: Double)	Agrega una página del tamaño indicado a un TsgcPDFDocument, dibuja la tabla desde AX, AY y agrega más páginas según sea necesario, repitiendo las filas de encabezado en cada página.

```
procedure CreateTablePDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oTable: TsgcPDFTable;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oTable := TsgcPDFTable.Create(3);
    try
      oTable.SetColumnWidths([200, 150, 145]);
      oTable.SetHeaders(['Product', 'Quantity', 'Price']);
      oTable.SetHeaderBackground(0.2, 0.4, 0.7);
      oTable.SetAlternateRowColors(0.93, 0.95, 1.0);
      oTable.SetBorder(0.5, 0.6, 0.6, 0.6);
      for i := 1 to 80 do
        oTable.AddRowData([AnsiString('Item ' + IntToStr(i)),
          AnsiString(IntToStr(i * 3)), AnsiString(Format('%d.00', [i * 10]))]);
      // rows that do not fit continue on new A4 pages
      oTable.DrawMultiPage(oPDF, 50, 790, 740, 595, 842);
    finally
      oTable.Free;
    end;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFDocument

Códigos de barras

TsgcPDFBarcodeGenerator (unidad sgcPDF_Barcodes) dibuja códigos de barras como gráficos vectoriales, de modo que se imprimen nítidos en cualquier tamaño. Simbologías admitidas (TsgcPDFBarcodeType): bcCode128, bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcQRCode, bcDataMatrix, bcPDF417 y bcGS1_128. Las versiones y niveles de corrección de errores de QR Code, DataMatrix, PDF417 y GS1-128 se describen en Códigos de barras 2D y GS1-128.
El primer parámetro de los métodos Draw es la página (TsgcPDFPage) o su **generador de contenido**, `TsgcPDFPage.Content`. Cualquier otro objeto genera una excepción. AX y AY son la esquina inferior izquierda.

Método	Descripción
DrawBarcode(APage: TObject; ABarcodeType; const AData: AnsiString; AX, AY: Double)	Dibuja con las opciones predeterminadas.
DrawBarcode(APage: TObject; ABarcodeType; const AData: AnsiString; AX, AY: Double; const AOptions: TsgcPDFBarcodeOptions)	Dibuja con opciones personalizadas.
DrawQRCode(APage: TObject; const AData: AnsiString; AX, AY, AModuleSize: Double)	Dibuja un QR Code cuyos módulos tienen un ancho de AModuleSize puntos.
EncodeLinear(ABarcodeType: TsgcPDFBarcodeType; const AData: AnsiString): TBytes	Devuelve los anchos de barras y espacios en módulos de una simbología lineal (Code 128, Code 39, EAN-13, EAN-8, UPC-A, ITF, Codabar y GS1-128), empezando por una barra: los mismos anchos que dibuja DrawBarcode. Los tipos 2D lanzan una excepción, tienen sus propios codificadores, véase Códigos de barras 2D y GS1-128.

TsgcPDFBarcodeOptions

Obtenga los valores predeterminados con `sgcPDFDefaultBarcodeOptions`: BarWidth 1.0, Height 50, ShowText True, TextFont sfHelvetica, TextFontSize 8, QuietZone 10, primer plano negro y fondo blanco (ForegroundR/G/B, BackgroundR/G/B).

```
procedure CreateBarcodesPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  oBarcodes: TsgcPDFBarcodeGenerator;
  oOptions: TsgcPDFBarcodeOptions;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    // barcodes are drawn into the page content builder
    oBarcodes := TsgcPDFBarcodeGenerator.Create;
    try
      oBarcodes.DrawBarcode(oPage.Content, bcCode128, 'SGC-2026-0001', 50, 700);
      oOptions := sgcPDFDefaultBarcodeOptions;
      oOptions.Height := 60;
      oOptions.BarWidth := 1.2;
      oBarcodes.DrawBarcode(oPage.Content, bcEAN13, '5901234123457', 50, 580, oOptions);
      // QR code, 3 points per module
      oBarcodes.DrawQRCode(oPage.Content, 'https://www.esegece.com', 350, 580, 3);
    finally
      oBarcodes.Free;
    end;
```

```
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Controles de código de barras

Para mostrar un código de barras en un formulario, coloque un TsgcPDFBarcodeControl (VCL) o un TsgcPDFBarcodeFMX (FireMonkey) y establezca Symbology y Text. Ambos controles admiten las once simbologías, incluidos los símbolos 2D y GS1-128 descritos en Códigos de barras 2D y GS1-128, y dibujan cada módulo en píxeles enteros. Alojan un TsgcPDFBarcodeSymbol, que también funciona sin formulario, por ejemplo en un servicio o en Linux:

- SaveToPNG y SaveToSVG guardan el símbolo como imagen.
- DrawToPDF dibuja el símbolo en una página PDF como rectángulos vectoriales, con los mismos módulos, giro, zona de silencio y línea de texto que el control.
- Runs, ModuleAt y LayoutAxes devuelven los módulos y la disposición en píxeles para el dibujo personalizado.

Vea también

- TsgcPDFDocument
- Códigos de barras 2D y GS1-128
- TsgcPDFBarcodeControl
- TsgcPDFBarcodeSymbol

Anotaciones

Las anotaciones se gestionan mediante `TsgcPDFDocument.Annotations` (TsgcPDFAnnotationManager, unidad sgcPDF_Annotations). Cada método Add toma el índice de página con base cero y devuelve la nueva anotación para que pueda establecer su color (SetColor), Author, Contents, Opacity y Flags.

Método	Anotación
AddText(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AText: WideString)	Nota adhesiva (TsgcPDFTextAnnotation, con Icon y Open).
AddLink(APageIndex; ALeft, ABottom, ARight, ATop: Double; ADestPage: Integer)	Enlace a otra página.
AddLinkURI(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AURI: AnsiString)	Enlace a una dirección web.
AddFreeText(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AText: WideString)	Cuadro de texto (SetFont, SetTextColor, Alignment).
AddHighlight / AddUnderline / AddStrikeOut(APageIndex; const AQuadPoints: array of Double)	Marcado de texto. Cada grupo de 8 números es un cuadrilátero.
AddStamp(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AStampName: AnsiString)	Sello de goma, como Approved o Draft.
AddLine(APageIndex; AX1, AY1, AX2, AY2: Double)	Línea con StartStyle y EndStyle (por ejemplo OpenArrow).
AddInk(APageIndex; const APoints: array of Double)	Dibujo a mano alzada a partir de pares X, Y.
AddSquare / AddCircle(APageIndex; ALeft, ABottom, ARight, ATop: Double)	Rectángulo o elipse (TsgcPDFShapeAnnotation, con SetInteriorColor).
AddReply(AParent: TsgcPDFAnnotation; const AText: WideString)	Anotación de texto que responde a AParent.

Cuando se carga un documento, sus anotaciones existentes se leen en el mismo administrador como objetos tipados, de modo que `Count`, `GetAnnotationsForPage` y `Annotations` las enumeran. Se pueden modificar y volver a guardar, vea Lectura de anotaciones y Edición y guardado más abajo.

```
procedure CreateAnnotationsPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oNote: TsgcPDFTextAnnotation;
  oHighlight: TsgcPDFHighlightAnnotation;
  oLine: TsgcPDFLineAnnotation;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.AddPage(595, 842).DrawText('Annotated page', 70, 790, sfHelveticaBold, 20);
    // sticky note: page index, then Left, Bottom, Right, Top
    oNote := oPDF.Annotations.AddText(0, 70, 680, 94, 704, 'Please review');
    oNote.SetColor(1, 1, 0);
    oNote.Icon := 'Comment';
    oNote.Author := 'Reviewer';
    // web link
    oPDF.Annotations.AddLinkURI(0, 70, 625, 280, 645, 'https://www.esegece.com');
    // highlight: one quadrilateral, 8 numbers (x1 y1 x2 y2 x3 y3 x4 y4)
```

```
    oHighlight := oPDF.Annotations.AddHighlight(0,
      [70, 786, 240, 786, 70, 806, 240, 806]);
    oHighlight.SetColor(1, 1, 0);
    oHighlight.Opacity := 0.5;
    oPDF.Annotations.AddStamp(0, 350, 420, 530, 470, 'Approved');
    oLine := oPDF.Annotations.AddLine(0, 70, 350, 300, 350);
    oLine.SetColor(1, 0, 0);
    oLine.EndStyle := 'OpenArrow';
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Formas

`AddSquare` y `AddCircle` añaden un rectángulo o una elipse inscrita en el rectángulo indicado, y devuelven una **TsgcPDFShapeAnnotation**. SetColor establece el color del borde, SetInteriorColor el color de relleno (no hay relleno si no se llama) y BorderWidth el ancho del borde en puntos. Las formas se guardan con un flujo de apariencia, de modo que todos los visores las dibujan igual.

```
procedure CreateShapesPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oBox, oCircle: TsgcPDFShapeAnnotation;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.AddPage(595, 842);
    // rectangle: page index, then Left, Bottom, Right, Top
    oBox := oPDF.Annotations.AddSquare(0, 70, 600, 270, 700);
    oBox.SetColor(0, 0, 1);
    oBox.SetInteriorColor(0.85, 0.9, 1);
    oBox.BorderWidth := 2;
    oBox.Contents := 'Check this block';
    // ellipse inscribed in the rectangle
    oCircle := oPDF.Annotations.AddCircle(0, 320, 600, 420, 700);
    oCircle.SetColor(1, 0, 0);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Lectura de anotaciones

LoadFromFile y LoadFromStream leen las anotaciones de todas las páginas en `Annotations`. Cada una se crea como la clase de su tipo, con sus propiedades rellenadas desde el archivo, de modo que la prueba `is` habitual o una conversión de tipo da acceso a ellas:

Subtipo en el archivo	Clase y propiedades leídas
Text	TsgcPDFTextAnnotation: Icon, Open.
Link	TsgcPDFLinkAnnotation: DestPage, o la dirección web.
FreeText	TsgcPDFFreeTextAnnotation: fuente, color del texto, Alignment.
Highlight, Underline, StrikeOut	TsgcPDFHighlightAnnotation (AnnotType indica cuál): QuadCount, QuadPoint.
Stamp	TsgcPDFStampAnnotation: StampName.

Subtipo en el archivo	Clase y propiedades leídas
Line	TsgcPDFLineAnnotation: X1, Y1, X2, Y2, StartStyle, EndStyle.
Ink	TsgcPDFInkAnnotation: PathCount, PathPointCount, GetPathPoint.
Square, Circle	TsgcPDFShapeAnnotation: HasInteriorColor, InteriorR, InteriorG, InteriorB.
Popup, FileAttachment, Widget and others	TsgcPDFAnnotation, la clase base.

Toda clase de anotación tiene estos miembros, tanto para las anotaciones cargadas desde un archivo como para las nuevas:

Miembro	Descripción
Author, Contents: WideString	Autor (/T) y texto (/Contents). El texto guardado como UTF-16, como lo escriben Acrobat y otros visores para los comentarios con acentos o en alfabetos no latinos, se decodifica a Unicode.
ModDate: TDateTime	Fecha de modificación (/M) como hora local, 0 cuando no hay.
Name: AnsiString	Nombre de la anotación (/NM).
InReplyTo: TsgcPDFAnnotation	La anotación a la que esta responde, o con la que está agrupada (/IRT). nil para una anotación de nivel superior.
ReplyType: AnsiString	Vacío o R para una respuesta, Group para un miembro de un grupo (/RT).
IsReply: Boolean	True cuando InReplyTo está establecido y ReplyType no es Group.
BorderWidth: Double	Ancho del borde o de la línea, en puntos.
HasColor, ColorR, ColorG, ColorB	El color (/C), tal como se lee del archivo o se establece con SetColor. Los componentes van de 0 a 1.
Modified: Boolean	True cuando una anotación cargada ha cambiado desde que se cargó el documento, o desde el último ResetModified. Un indicador para la interfaz de usuario. Siempre False para una anotación creada en código.
ChangedSinceLoad: Boolean	True cuando una anotación cargada difiere del estado que tenía al cargar el documento, de modo que el siguiente guardado la escribe de nuevo. ResetModified no lo modifica. Siempre False para una anotación creada en código.
MoveBy(ADX, ADY: Double)	Mueve la anotación ADX, ADY puntos: su Rect y los puntos de la línea, los puntos de los cuadriláteros o los trazos de tinta.
SourceObjectNumber: Integer	Número de objeto de una anotación cargada en el archivo, 0 para una creada en código.

La geometría de líneas, marcas de texto y tinta se lee con X1, Y1, X2 e Y2 (TsgcPDFLineAnnotation), QuadCount y QuadPoint(AIndex) con AIndex de 0 a QuadCount * 8 - 1 (TsgcPDFHighlightAnnotation), y PathCount, PathPointCount(APath) y GetPathPoint(APath, AIndex, AX, AY) (TsgcPDFInkAnnotation).

```
procedure ListAnnotations(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oAnnot: TsgcPDFAnnotation;
  oReplies: TList;
```

```
  i, j: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    for i := 0 to oPDF.Annotations.Count - 1 do
    begin
      oAnnot := TsgcPDFAnnotation(oPDF.Annotations.Annotations[i]);
      // replies are listed under their parent, popups have no content
      if oAnnot.IsReply or (oAnnot.AnnotType = atPopup) then
        Continue;
      Write(Format('Page %d: %s', [oAnnot.PageIndex + 1, oAnnot.ClassName]));
      if oAnnot.Author <> '' then
        Write(', by ', oAnnot.Author);
      if oAnnot.ModDate <> 0 then
        Write(', ', DateTimeToStr(oAnnot.ModDate));
      WriteLn;
      if oAnnot.Contents <> '' then
        WriteLn('  ', oAnnot.Contents);
      if oAnnot is TsgcPDFStampAnnotation then
        WriteLn('  stamp: ', TsgcPDFStampAnnotation(oAnnot).StampName);
      oReplies := oPDF.Annotations.GetReplies(oAnnot);
      try
        for j := 0 to oReplies.Count - 1 do
          WriteLn('  reply from ', TsgcPDFAnnotation(oReplies[j]).Author, ': ',
            TsgcPDFAnnotation(oReplies[j]).Contents);
      finally
        oReplies.Free;
      end;
    end;
  finally
    oPDF.Free;
  end;
end;

procedure WriteGeometry(AAnnot: TsgcPDFAnnotation);
var
  oLine: TsgcPDFLineAnnotation;
  oMarkup: TsgcPDFHighlightAnnotation;
  oInk: TsgcPDFInkAnnotation;
  i: Integer;
  vX, vY: Double;
begin
  if AAnnot is TsgcPDFLineAnnotation then
  begin
    oLine := TsgcPDFLineAnnotation(AAnnot);
    WriteLn(Format('Line from %.1f, %.1f to %.1f, %.1f',
      [oLine.X1, oLine.Y1, oLine.X2, oLine.Y2]));
  end
  else if AAnnot is TsgcPDFHighlightAnnotation then
  begin
    // highlight, underline or strike-out: 8 numbers per quadrilateral
    oMarkup := TsgcPDFHighlightAnnotation(AAnnot);
    for i := 0 to oMarkup.QuadCount - 1 do
      WriteLn(Format('Quad %d: first corner %.1f, %.1f',
        [i, oMarkup.QuadPoint(i * 8), oMarkup.QuadPoint(i * 8 + 1)]));
  end
  else if AAnnot is TsgcPDFInkAnnotation then
  begin
    oInk := TsgcPDFInkAnnotation(AAnnot);
    for i := 0 to oInk.PathCount - 1 do
    begin
      oInk.GetPathPoint(i, 0, vX, vY);
      WriteLn(Format('Path %d: %d points, starts at %.1f, %.1f',
        [i, oInk.PathPointCount(i), vX, vY]));
    end;
  end;
```

```
  if AAnnot.HasColor then
    WriteLn(Format('Color %.2f %.2f %.2f, border %.1f',
      [AAnnot.ColorR, AAnnot.ColorG, AAnnot.ColorB, AAnnot.BorderWidth]));
end;
```

Respuestas

Una respuesta es una anotación cuyo `InReplyTo` apunta a otra anotación (la entrada /IRT del archivo), como las respuestas a un comentario en Acrobat. LoadFromFile enlaza cada respuesta con su padre, también cuando el padre está en otra página. El administrador tiene estos miembros para las respuestas y para la propia lista:

Miembro	Descripción
GetAnnotationsForPage(APageIndex: Integer): TList	Las anotaciones de una página, en el orden de la lista. Libere la lista.
GetReplies(AAnnotation: TsgcPDFAnnotation): TList	Las respuestas directas de una anotación, en el orden de la lista. Libere la lista.
AddReply(AParent: TsgcPDFAnnotation; const AText: WideString): TsgcPDFTextAnnotation	Añade una anotación de texto que responde a AParent, en su página y con su Rect.
Remove(AAnnotation: TsgcPDFAnnotation)	Elimina y libera una anotación, todas las anotaciones que responden a ella o están agrupadas con ella (de forma recursiva) y su popup.
LinkReplies	Resuelve el /IRT de las anotaciones cargadas a la anotación cargada con ese número de objeto. LoadFromFile lo llama; llámelo usted mismo solo después de LoadFromPage.
Modified: Boolean	True cuando se ha añadido o eliminado una anotación, o una anotación cargada ha cambiado, desde que se cargó el documento o desde el último ResetModified. El indicador de "cambios sin guardar" para la interfaz de usuario.
ChangedSinceLoad: Boolean	True cuando se ha añadido o eliminado una anotación, o una anotación cargada difiere del archivo, desde que se cargó el documento. ResetModified no lo modifica.
ResetModified	Borra Modified, por ejemplo después de guardar. Las ediciones se conservan, y el siguiente guardado todavía las escribe. Vea Seguimiento de cambios.
MarkLoaded	Toma el estado actual como el estado del archivo y borra Modified y ChangedSinceLoad. El documento lo llama después de cargar. Vea Seguimiento de cambios.

Las respuestas se escriben con /IRT apuntando a su padre, tanto con SaveToFile como con SaveIncremental, de modo que sgcPDF y otros visores las leen como respuestas. AddReply no establece Author ni ModDate, establézcalos usted mismo.

Edición y guardado

Las propiedades de una anotación cargada se pueden modificar igual que las de una nueva: Contents, Author, Rect, el color, Flags, ModDate, etcétera, o MoveBy. SaveToFile y SaveIncremental escriben los cambios.

- SaveIncremental vuelve a escribir una anotación modificada con su propio número de objeto, después de los bytes originales del archivo. Las anotaciones y respuestas nuevas se añaden a su página en la misma revisión.
- Cuando su código no ha cambiado ModDate, se establece a la hora actual.
- El flujo de apariencia del archivo se conserva cuando solo cambian Contents, Author, Name, ModDate, Rect, Flags o las entradas de respuesta. Con otros cambios, los sellos y las formas reciben una apariencia nueva, y los demás tipos pierden la suya para que los visores los dibujen de nuevo a partir de sus propiedades.
- Remove elimina una anotación con sus respuestas y su popup, y el siguiente guardado escribe la página sin ellos.

Como SaveIncremental no modifica los bytes originales, el rango de bytes de cada firma existente sigue siendo el mismo y el valor de la firma sigue siendo válido. Que un validador acepte o no la nueva revisión depende de los permisos de la firma. Una firma de certificación con permiso DocMDP P=3, o solo firmas de aprobación, permiten añadir y modificar comentarios. Con P=1 o P=2 el validador informa de un cambio no permitido. Un guardado completo con SaveToFile escribe de nuevo todo el archivo e invalida las firmas existentes, así que úselo solo en documentos sin firmar. Vea Guardado incremental y Firmas digitales.

```
procedure ReviewNotes(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oList: TList;
  oAnnot: TsgcPDFAnnotation;
  oReply: TsgcPDFTextAnnotation;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oList := oPDF.Annotations.GetAnnotationsForPage(0);
    try
      for i := 0 to oList.Count - 1 do
      begin
        oAnnot := TsgcPDFAnnotation(oList[i]);
        if (oAnnot is TsgcPDFTextAnnotation) and not oAnnot.IsReply then
        begin
          // change a loaded note
          oAnnot.Contents := oAnnot.Contents + ' (checked)';
          // and answer it
          oReply := oPDF.Annotations.AddReply(oAnnot, 'Fixed in version 2');
          oReply.Author := 'Reviewer';
          oReply.ModDate := Now;
          Break;
        end;
      end;
    finally
      oList.Free;
    end;
    // appends the changes after the original bytes of the file
    if oPDF.Annotations.Modified then
      oPDF.SaveIncremental(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Seguimiento de cambios

Dos indicadores dicen qué ha cambiado, y cada uno tiene su función.

- `Modified` es el indicador de "cambios sin guardar" de la interfaz de usuario. `Annotations.Modified` es True cuando se ha añadido o eliminado una anotación, o una anotación cargada ha cambiado, desde que se cargó el documento o desde el último ResetModified. La propiedad `Modified` de cada anotación indica qué anotaciones cargadas han cambiado. Úselos para habilitar un botón Guardar, o para preguntar al usuario antes de cerrar.
- `ChangedSinceLoad` compara con el archivo. `Annotations.ChangedSinceLoad` es True cuando se ha añadido o eliminado una anotación, o una anotación cargada difiere del estado que tenía al cargar el documento. Save-

ToFile y SaveIncremental usan la función `ChangedSinceLoad` de cada anotación para decidir qué anotaciones cargadas escriben con sus cambios.

`ResetModified` solo borra Modified. Llámelo después de guardar, para que la interfaz de usuario muestre que no queda nada por guardar. No modifica ChangedSinceLoad, así que nunca pierde una edición: el siguiente SaveToFile o SaveIncremental todavía escribe todos los cambios hechos desde que se cargó el documento, junto con los más recientes. No hace falta volver a cargar el archivo guardado, y el ejemplo sigue trabajando con el mismo documento.

`MarkLoaded` toma el estado actual de las anotaciones como el estado del archivo y borra los dos indicadores. LoadFromFile y LoadFromStream lo llaman después de leer las anotaciones, de modo que las aplicaciones rara vez lo necesitan. Para borrar el indicador de "cambios sin guardar" después de guardar, llame a ResetModified.

```
procedure MoveNoteAndSave(ADocument: TsgcPDFDocument; const AFileName: string);
var
  oAnnot: TsgcPDFAnnotation;
  i: Integer;
begin
  for i := 0 to ADocument.Annotations.Count - 1 do
  begin
    oAnnot := TsgcPDFAnnotation(ADocument.Annotations.Annotations[i]);
    if oAnnot is TsgcPDFTextAnnotation then
    begin
      // 20 points right and 20 points down
      oAnnot.MoveBy(20, -20);
      WriteLn('Note changed: ', oAnnot.Modified);
      Break;
    end;
  end;
  if ADocument.Annotations.Modified then
  begin
    ADocument.SaveIncremental(AFileName);
    // saved: clear the "unsaved changes" flag of the user interface
    ADocument.Annotations.ResetModified;
  end;
  WriteLn('Unsaved changes: ', ADocument.Annotations.Modified);
  // still True: the next save writes the note again, with the later changes
  WriteLn('Changed since load: ', ADocument.Annotations.ChangedSinceLoad);
end;
```

Renderizado de anotaciones editadas

Para uso avanzado, `TsgcPDFDocument.BuildPageAnnotations(APageIndex)` devuelve las anotaciones de una página tal como están ahora, con las ediciones incluidas, como un TsgcPDFArray para la propiedad Annotations de un renderizador. Quien llama libera el array. Los visores lo usan para dibujar las páginas cuyas anotaciones se han modificado, así que la mayoría de las aplicaciones nunca lo llaman.

Vea también

- TsgcPDFDocument
- Marcadores
- Guardado incremental
- TsgcPDFViewerControl
- TsgcPDFViewerFMX
- TsgcPDFCommentsView

Marcadores

Los marcadores (el esquema del documento) se gestionan mediante `TsgcPDFDocument.Bookmarks` (TsgcPDFBookmarks, unidad sgcPDF_Bookmarks). `Add` crea una entrada de nivel superior y `TsgcPDFBookmarkItem.AddChild` una anidada. Cada elemento tiene Title, PageIndex, Top, Left, Zoom, ColorR/G/B, Bold, Italic y Open.

```
procedure CreateBookmarksPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oChapter: TsgcPDFBookmarkItem;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    for i := 0 to 4 do
      oPDF.AddPage(595, 842).DrawText(AnsiString('Page ' + IntToStr(i + 1)),
        50, 780, sfHelvetica, 14);
    oPDF.Bookmarks.Add('Cover', 0);
    oChapter := oPDF.Bookmarks.Add('Chapter 1', 1);
    oChapter.Bold := True;
    oChapter.Open := True;
    oChapter.AddChild('Section 1.1', 2);
    oChapter.AddChild('Section 1.2', 3);
    oPDF.Bookmarks.Add('Appendix', 4);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Lectura de marcadores

Los marcadores de un documento cargado están disponibles a través de la misma propiedad.

```
procedure PrintBookmarks(AItem: TsgcPDFBookmarkItem; const AIndent: string);
var
  i: Integer;
begin
  WriteLn(AIndent, string(AItem.Title), ' -> page ', AItem.PageIndex + 1);
  for i := 0 to AItem.ChildCount - 1 do
    PrintBookmarks(AItem.GetChild(i), AIndent + '  ');
end;

procedure ListBookmarks(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    for i := 0 to oPDF.Bookmarks.Count - 1 do
      PrintBookmarks(oPDF.Bookmarks[i], '');
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFDocument

- Anotaciones

Capas

Las capas (grupos de contenido opcional) permiten que el lector muestre u oculte partes de una página. Se gestionan mediante `TsgcPDFDocument.Layers` (TsgcPDFLayerManager, unidad sgcPDF_Layers).

- `AddLayer(AName, ADefaultState)` crea una capa que inicialmente está activada (lsOn) o desactivada (lsOff). TsgcPDFLayer también tiene Intent, Visible, Printable y Locked.
- El contenido pertenece a una capa cuando se escribe entre `BeginLayerMarkedContent(ALayer)` y `EndLayerMarkedContent`. Ambos devuelven el texto del operador, que usted agrega a la página con `Content.AddOperator`.
- `AddGroup` agrupa capas en el panel de capas del lector. `FindLayer`, `GetLayer`, `LayerCount` y `RemoveLayer` gestionan la lista, que también se rellena al cargar un documento.

```
procedure CreateLayersPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  oNotes: TsgcPDFLayer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.Layers.AddLayer('Background', lsOn);
    oNotes := oPDF.Layers.AddLayer('Notes', lsOff);
    oPage := oPDF.AddPage(595, 842);
    oPage.DrawText('Always visible', 50, 780, sfHelvetica, 14);
    // wrap the drawing operators that belong to the layer in marked content
    oPage.Content.AddOperator(oPDF.Layers.BeginLayerMarkedContent(oNotes));
    oPage.DrawText('Only visible when the Notes layer is on', 50, 740,
      sfHelvetica, 14);
    oPage.Content.AddOperator(oPDF.Layers.EndLayerMarkedContent);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFDocument
- Gráficos

Formularios

Los formularios interactivos (AcroForms) se gestionan mediante `TsgcPDFDocument.AcroForm` (TsgcPDFAcroForm, unidad sgcPDF_Forms).

Creación de campos

Método	Campo
AddTextField(const AName: AnsiString; APageIndex: Integer; ALeft, ABottom, ARight, ATop: Double)	Cuadro de texto.
AddCheckBox(...)	Casilla de verificación.
AddRadioGroup(const AName: AnsiString) and AddRadioButton(AGroup; APageIndex; ALeft, ABottom, ARight, ATop; const AExportValue)	Botones de opción.
AddComboBox(...; const AOptions: array of WideString)	Lista desplegable.
AddListBox(...; const AOptions: array of WideString)	Lista.
AddButton(...; const ACaption: WideString)	Botón.
AddSignatureField(...)	Campo de firma vacío.

Cada método devuelve un TsgcPDFFormField con Value, DefaultValue, ReadOnly, Required, MaxLen, FontName, FontSize y Alignment. Establezca `NeedAppearances` en True para que los lectores regeneren las apariencias de los campos.

```
procedure CreateFormPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    oPage.DrawText('Name:', 50, 737, sfHelvetica, 12);
    oPage.DrawText('Country:', 50, 697, sfHelvetica, 12);
    oPage.DrawText('Subscribe:', 50, 657, sfHelvetica, 12);
    oPDF.AcroForm.AddTextField('name', 0, 160, 730, 400, 755).FontSize := 12;
    oPDF.AcroForm.AddComboBox('country', 0, 160, 690, 400, 715,
      ['Germany', 'Spain', 'United States']);
    oPDF.AcroForm.AddCheckBox('subscribe', 0, 160, 650, 178, 668);
    oPDF.AcroForm.NeedAppearances := True;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Relleno de campos

Cuando se carga un documento, sus campos se leen en AcroForm. Utilice FillField, SetCheckBox, GetFieldValue, FindField, GetField y FieldCount.

```
procedure FillFormPDF(const AInput, AOutput: string);
var
  oPDF: TsgcPDFDocument;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AInput);
    for i := 0 to oPDF.AcroForm.FieldCount - 1 do
      WriteLn('Field: ', string(oPDF.AcroForm.GetField(i).FullName));
    oPDF.AcroForm.FillField('name', 'John Smith');
    oPDF.AcroForm.FillField('country', 'Spain');
    oPDF.AcroForm.SetCheckBox('subscribe', True);
    oPDF.SaveToFile(AOutput);
  finally
    oPDF.Free;
  end;
end;
```

Aplanado

El aplanado dibuja los valores actuales de los campos como contenido de página normal y elimina los campos, de modo que los valores ya no se pueden editar. Utilice `TsgcPDFAcroForm.FlattenFields` o `TsgcPDFEditor.FlattenForm`.

```
procedure FillAndFlatten(const AInput, AOutput: string);
var
  oEditor: TsgcPDFEditor;
begin
  oEditor := TsgcPDFEditor.Create;
  try
    oEditor.LoadFromFile(AInput);
    oEditor.FillFormField('name', 'Jane Doe');
    // the field values become ordinary page content, the fields are removed
    oEditor.FlattenForm;
    oEditor.SaveToFile(AOutput);
  finally
    oEditor.Free;
  end;
end;
```

Vea también

- TsgcPDFDocument
- TsgcPDFEditor

Cifrado

Llame a `SetEncryption` antes de guardar para cifrar un documento con el manejador de seguridad estándar. **TsgcPDFEncryptionMethod** (unidad sgcPDF_Encryption) selecciona el algoritmo:

Valor	Algoritmo
emRC4_40	RC4 de 40 bits (PDF 1.1, solo heredado)
emRC4_128	RC4 de 128 bits (PDF 1.4)
emAES_128	AES-128 (PDF 1.6)
emAES_256	AES-256 (PDF 2.0, recomendado)

La contraseña de usuario es necesaria para abrir el archivo. La contraseña de propietario otorga acceso completo. SetEncryption en TsgcPDFDocument y TsgcPDFEditor otorga todos los permisos (imprimir, modificar, copiar, anotar, rellenar formularios, extraer, ensamblar e impresión de alta calidad).

```
procedure CreateEncryptedPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.AddPage(595, 842).DrawText('Secret', 50, 780, sfHelveticaBold, 24);
    // user password opens the file, owner password grants full rights
    oPDF.SetEncryption(emAES_256, 'user123', 'owner456');
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Apertura de archivos cifrados

Llame a `SetPassword` antes de `LoadFromFile`. Tanto la contraseña de usuario como la de propietario abren el archivo.

```
procedure OpenEncryptedPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // set the password before loading
    oPDF.SetPassword('user123');
    oPDF.LoadFromFile(AFileName);
    WriteLn(string(oPDF.ExtractTextW(0)));
  finally
    oPDF.Free;
  end;
end;
```

Acceso de bajo nivel

TsgcPDFEncryptionHandler implementa el propio manejador de seguridad: `SetupEncryption` con un conjunto TsgcPDFPermissions, `Authenticate`, y las rutinas de cifrado y descifrado para cadenas y streams. La unidad tam-

bién expone sgcPDFMD5, sgcPDFSHA256, sgcPDFSHA384, sgcPDFSHA512, sgcPDFRC4, sgcPDFAESEncryptCBC y sgcPDFAESDecryptCBC. Los errores de esta unidad generan **EsgcPDFEncryptionError**.

Vea también

- TsgcPDFDocument
- Firmas digitales

Firmas digitales

TsgcPDFSignatureHandler firma un PDF con un certificado que tiene una clave privada, cargada desde un archivo PFX, desde el almacén de certificados de Windows por nombre de asunto, o por huella digital. La firma es una firma PKCS#7 separada (adbe.pkcs7.detached) añadida como una actualización incremental, de modo que los bytes del archivo original se mantienen sin cambios al comienzo del archivo firmado.

Firma

```
procedure SignPDF(const AInput, AOutput: string);
var
  oSigner: TsgcPDFSignatureHandler;
begin
  oSigner := TsgcPDFSignatureHandler.Create;
  try
    oSigner.LoadCertificateFromPFX('signer.pfx', 'test123');
    WriteLn('Signing as ', string(oSigner.GetCertificateInfo.SubjectName));
    oSigner.Reason := 'Document approval';
    oSigner.Location := 'Barcelona';
    oSigner.HashAlgorithm := shaSHA256;
    // visible signature field on page 1: Left, Bottom, Right, Top
    oSigner.SetField('Signature1', 0, 50, 50, 250, 100);
    oSigner.SignDocument(AInput, AOutput);
  finally
    oSigner.Free;
  end;
end;
```

Llame a `SetField` para un rectángulo de firma visible, o a `SetInvisible` para una firma sin apariencia. `HashAlgorithm` selecciona SHA-1, SHA-256 (predeterminado), SHA-384 o SHA-512.

Verificación

TsgcPDFSignatureVerifier informa, para cada firma, si es válida, si el documento se modificó después de firmarlo, y si el certificado es de confianza para Windows.

```
procedure VerifyPDF(const AFileName: string);
var
  oVerifier: TsgcPDFSignatureVerifier;
  oCert: TsgcPDFCertificateInfo;
  i: Integer;
begin
  oVerifier := TsgcPDFSignatureVerifier.Create;
  try
    oVerifier.LoadDocument(AFileName);
    for i := 0 to oVerifier.GetSignatureCount - 1 do
    begin
      oCert := oVerifier.GetSignatureInfo(i);
      WriteLn('Signer:   ', string(oCert.SubjectName));
      WriteLn('Valid:    ', oVerifier.VerifySignature(i));
      WriteLn('Modified: ', oVerifier.IsDocumentModified(i));
      WriteLn('Trusted:  ', oVerifier.IsSignatureTrusted(i));
    end;
  finally
    oVerifier.Free;
  end;
end;
```

Firma con el editor

`TsgcPDFEditor.SignDocument(AOutputFile, ACertSubjectName, AReason, ALocation)` guarda y firma el documento editado con un certificado del almacén de Windows en una sola llamada.

Vea también

- Cifrado
- PDF/A

PDF/A

PDF/A es el estándar ISO para el archivado a largo plazo. sgcPDF escribe los niveles de conformidad **b** (básico) de PDF/A-1, PDF/A-2 y PDF/A-3, seleccionados con **TsgcPDFALevel** (unidad sgcPDF_PDFA): paA1b, paA2b y paA3b.
Cuando se establece `SetPDFALevel`, el documento se guarda con metadatos XMP que declaran el nivel de conformidad y repiten los campos de Info, un intento de salida sRGB y un ID de documento. Su contenido también debe seguir las reglas de PDF/A:

- Incruste todas las fuentes: dibuje el texto con fuentes TrueType (AddTrueTypeFont y DrawTextTT), no con las fuentes estándar.
- No cifre el documento.
- PDF/A-1b no permite transparencia, así que evite SetOpacity y el dibujo transparente en ese nivel.

```
procedure CreatePDFA(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  vFont: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.SetPDFALevel(paA2b);
    oPDF.Info.Title := 'Archived invoice';
    oPDF.Info.Author := 'eSeGeCe';
    // PDF/A requires embedded fonts, so use a TrueType font
    vFont := oPDF.AddTrueTypeFont('Arial', False, False);
    oPDF.AddPage(595, 842).DrawTextTT('PDF/A-2b document', 50, 780, vFont, 20);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Validación

TsgcPDFAValidator comprueba un documento cargado con respecto a un nivel: metadatos XMP, incrustación de fuentes, espacios de color, transparencia, cifrado, anotaciones, acciones y estructura del archivo. `Errors` enumera los problemas encontrados. Es una comprobación rápida propia, no un sustituto de un validador de conformidad completo como veraPDF.

```
procedure ValidatePDFA(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oValidator: TsgcPDFAValidator;
begin
  oPDF := TsgcPDFDocument.Create;
  oValidator := TsgcPDFAValidator.Create(paA2b);
  try
    oPDF.LoadFromFile(AFileName);
    if oValidator.Validate(oPDF.Parser) then
      WriteLn('No PDF/A-2b problems found')
    else
      WriteLn(oValidator.Errors.Text);
  finally
    oValidator.Free;
    oPDF.Free;
  end;
end;
```

`TsgcPDFEditor.ConvertToPDFA(ALevel)` establece el nivel en un documento abierto con el editor.

Vea también

- TsgcPDFDocument
- Texto y fuentes

Optimización de archivos PDF

TsgcPDFOptimizer reescribe un PDF para reducir su tamaño. De forma predeterminada, comprime streams, combina objetos duplicados y descarta objetos no utilizados. La reducción de resolución de imágenes y la eliminación de metadatos, marcadores, anotaciones, JavaScript y miniaturas se activan mediante `Options`.

```
procedure OptimizePDF(const AInput, AOutput: string);
var
  oOptimizer: TsgcPDFOptimizer;
  oOptions: TsgcPDFOptimizeOptions;
begin
  oOptimizer := TsgcPDFOptimizer.Create;
  try
    oOptions := sgcPDFDefaultOptimizeOptions;
    oOptions.DownsampleImages := True;
    oOptions.MaxImageDPI := 150;
    oOptions.RemoveMetadata := True;
    oOptimizer.Options := oOptions;
    oOptimizer.Optimize(AInput, AOutput);
    WriteLn('Saved ', oOptimizer.GetSizeReduction, ' bytes');
    WriteLn(oOptimizer.GetLog.Text);
  finally
    oOptimizer.Free;
  end;
end;
```

Un archivo que ya es compacto, como un archivo escrito por el propio sgcPDF, puede no reducirse más. GetSizeReduction devuelve entonces 0 y el registro informa de los tamaños de entrada y salida.

Vea también

- Imágenes

Combinar, dividir y extraer páginas

Funciones auxiliares de una llamada

La unidad sgcPDF_Editor tiene funciones que operan directamente sobre archivos:

- `sgcPDFMergeFiles(AInputFiles, AOutputFile)` combina archivos en el orden indicado.
- `sgcPDFExtractPages(AInputFile, AOutputFile, AFromPage, AToPage)` copia un rango de páginas con base cero e inclusivo a un archivo nuevo (división).
- `sgcPDFGetPageCount(AFileName)` devuelve el número de páginas.

```
procedure MergeAndSplit;
begin
  // merge several files into one
  sgcPDFMergeFiles(['out\hello.pdf', 'out\bookmarks.pdf', 'out\tables.pdf'],
    'out\merged.pdf');
  WriteLn('Merged pages: ', sgcPDFGetPageCount('out\merged.pdf'));
  // extract pages 2 to 4 (zero based indexes 1 to 3) into a new file
  sgcPDFExtractPages('out\merged.pdf', 'out\pages_2_4.pdf', 1, 3);
end;
```

Con el editor

TsgcPDFEditor combina operaciones de página en un documento abierto: MergeFrom, ExtractPages, DeletePage, RotatePage y AddBlankPage.

```
procedure ReorganizePages(const AInput, AOutput: string);
var
  oEditor: TsgcPDFEditor;
begin
  oEditor := TsgcPDFEditor.Create;
  try
    oEditor.LoadFromFile(AInput);
    oEditor.MergeFrom('out\hello.pdf');
    oEditor.RotatePage(0, 90);
    oEditor.DeletePage(1);
    oEditor.AddBlankPage(595, 842);
    oEditor.SaveToFile(AOutput);
  finally
    oEditor.Free;
  end;
end;
```

Con un TsgcPDFDocument, utilice AppendDocument, InsertPage y DeletePage.

Vea también

- TsgcPDFEditor

Extracción de texto

sgcPDF extrae el texto de los streams de contenido de página. Decodifica las codificaciones estándar (WinAnsi, MacRoman, Standard), las Differences de fuente, los mapas ToUnicode y las fuentes CID, de modo que el texto escrito con fuentes TrueType o CJK incrustadas se devuelve como Unicode. Las páginas escaneadas que solo contienen imágenes no tienen texto que extraer.

Texto plano

`TsgcPDFDocument.ExtractTextW` devuelve el texto de una página y `FindText` lo busca.

```
procedure PrintAllText(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    for i := 0 to oPDF.PageCount - 1 do
    begin
      WriteLn('--- page ', i + 1, ' ---');
      WriteLn(string(oPDF.ExtractTextW(i)));
    end;
    if oPDF.FindText(0, 'Hello') then
      WriteLn('Page 1 contains "Hello"');
  finally
    oPDF.Free;
  end;
end;
```

Texto con posiciones

TsgcPDFTextExtractor devuelve cada fragmento de texto con sus coordenadas, fuente y tamaño.

```
procedure PrintTextPositions(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oExtractor: TsgcPDFTextExtractor;
  oFragment: TsgcPDFTextFragment;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oExtractor := TsgcPDFTextExtractor.Create(oPDF.Parser,
      oPDF.Pages[0].PageDict);
    try
      oExtractor.Extract;
      for i := 0 to oExtractor.GetFragmentCount - 1 do
      begin
        oFragment := oExtractor.GetFragment(i);
        WriteLn(Format('%.1f, %.1f  %s  (%s %.1f pt)', [oFragment.X, oFragment.Y,
          string(oFragment.Text), string(oFragment.FontName), oFragment.FontSize]));
      end;
    finally
      oExtractor.Free;
    end;
  finally
    oPDF.Free;
  end;
```

```
end;
```

Usar el analizador directamente

Las funciones sgcPDFExtractPageText, sgcPDFExtractAllText y sgcPDFFindText funcionan con un TsgcPDFParser. Llame a `Parse` después de cargar.

```
function GetDocumentText(const AFileName: string): WideString;
var
  oParser: TsgcPDFParser;
begin
  oParser := TsgcPDFParser.Create;
  try
    oParser.LoadFromFile(AFileName);
    oParser.Parse;
    Result := sgcPDFExtractAllText(oParser);
  finally
    oParser.Free;
  end;
end;
```

Vea también

- TsgcPDFTextExtractor

Renderizado y exportación

Las páginas se renderizan con **TsgcPDFRendererGDI** (unidad sgcPDF_Renderer_GDI), que dibuja el contenido de la página (trayectos, texto, imágenes, degradados) en un contexto de dispositivo de Windows. Rara vez lo utilizará directamente:

- TsgcPDFExporter guarda páginas como archivos BMP, PNG o JPEG, o miniaturas.
- TsgcPDFViewer muestra páginas en una ventana.
- `TsgcPDFRendererGDI.RenderPage(ADC, AParser, APageDict, AWidth, AHeight)` dibuja una página en cualquier HDC, por ejemplo un canvas de TBitmap o una impresora.

Exportación de páginas

```
procedure ExportPages(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oExporter: TsgcPDFExporter;
  oOptions: TsgcPDFExportOptions;
begin
  oPDF := TsgcPDFDocument.Create;
  oExporter := TsgcPDFExporter.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oOptions := sgcPDFDefaultExportOptions;
    oOptions.Format := efPNG;
    oOptions.DPI := 150;
    oExporter.ExportPage(oPDF, 0, 'out\page1.png', oOptions);
    oExporter.ExportThumbnail(oPDF, 0, 'out\thumb1.bmp', 200, 200);
  finally
    oExporter.Free;
    oPDF.Free;
  end;
  // one call helpers that open the file themselves
  sgcPDFPageToImage(AFileName, 'out\page1_96.png', 0, 96);
end;
```

Visualización

```
procedure ShowViewer(const AFileName: string);
var
  oViewer: TsgcPDFViewer;
begin
  oViewer := TsgcPDFViewer.Create;
  try
    oViewer.ViewMode := vmContinuous;
    // top level window; use CreateWindow(AParent, X, Y, W, H) to embed it
    oViewer.CreateWindow(100, 100, 800, 900, 'My PDF Viewer');
    oViewer.LoadFromFile(AFileName);
    oViewer.ZoomFit;
    oViewer.Show;
    oViewer.RunMessageLoop;
  finally
    oViewer.Free;
  end;
end;
```

Vea también

- TsgcPDFExporter
- TsgcPDFViewer

Gestión de errores y entradas hostiles

sgcPDF informa de los errores generando excepciones. Cargar un archivo dañado o no admitido, un índice de página incorrecto, un certificado ausente o una imagen no admitida genera una excepción cuyo Message describe el problema (por ejemplo *PDF parse error: cannot find startxref* o *sgcPDF: Invalid page index*). Los errores de cifrado generan **EsgcPDFEncryptionError**. Envuelva en try..except la carga de archivos que usted no controla.

```
procedure OpenUntrustedPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  // cap every decoded stream at 64 MB instead of the default 256 MB
  sgcPDFMaxDecodedSize := 64 * 1024 * 1024;
  oPDF := TsgcPDFDocument.Create;
  try
    try
      oPDF.LoadFromFile(AFileName);
      WriteLn('Pages: ', oPDF.PageCount);
    except
      on E: Exception do
        WriteLn('Rejected: ', E.Message);
    end;
  finally
    oPDF.Free;
  end;
end;
```

Límites para archivos no confiables

Los archivos PDF se pueden fabricar para agotar la memoria o la pila, por ejemplo con un stream comprimido pequeño que se descomprime en gigabytes, o con objetos anidados profundamente. El analizador y los decodificadores aplican límites y generan una excepción cuando se supera alguno:

Límite	Valor	Descripción
sgcPDFMaxDecodedSize	268435456 (256 MB)	Tamaño decodificado máximo de un stream (Flate, LZW, ASCII85, ASCIIHex, RunLength). Es una variable global en sgcPDF_Filters, por lo que puede reducirla para servicios que procesan cargas, o aumentarla para imágenes legítimas muy grandes.
Anidamiento de objetos	512 niveles	Arrays y diccionarios anidados en el analizador.
Número de objetos	8388607	Mayor número de objeto aceptado a partir de los datos de referencias cruzadas.
Secciones de referencias cruzadas	4096	Secciones y streams xref encadenados.
Tamaño de imagen	2^{27} píxeles, 1 GB de datos	Mayor imagen que se decodifica.
Anidamiento de streams de contenido	64 niveles	Profundidad de anidamiento aceptada por el analizador del stream de contenido.

Límite	Valor	Descripción
Árbol de páginas	64 niveles, 1000000 nodos	Profundidad y tamaño del árbol de páginas recorrido por el extractor de texto.

Solo se puede cambiar sgcPDFMaxDecodedSize. Los demás límites son fijos. Establezca sgcPDFMaxDecodedSize una sola vez al iniciar, porque se aplica a todos los documentos del proceso.

Vea también

- Preguntas frecuentes

Arquitectura del renderizador

sgcPDF dibuja páginas PDF con un único intérprete de contenido y varios backends de dibujo. El intérprete lee el stream de contenido de la página, resuelve fuentes, imágenes, espacios de color, degradados y anotaciones, y transforma cada coordenada a píxeles de dispositivo. El backend solo tiene que pintar trayectos, rásteres y glifos. Por lo tanto, la misma página se ve igual en todos los backends.

Clase	Unidad	Función
TsgcPDFRenderer	sgcPDF_Renderer	El intérprete de contenido. Independiente de la plataforma, en Pascal puro.
TsgcPDFRenderCanvas	sgcPDF_Renderer_Canvas	Backend de dibujo abstracto: trayectos, recorte, imágenes, rásteres, capas y texto del sistema.
TsgcPDFRasterCanvas	sgcPDF_Renderer_Raster	Rasterizador por software con antialiasing. Dibuja en un búfer de píxeles BGRA. Se ejecuta en todas las plataformas.
TsgcPDFGDICanvas	sgcPDF_Renderer_GDI	Backend GDI de Windows. Dibuja en cualquier contexto de dispositivo (pantalla, mapa de bits, impresora).
TsgcPDFFMXCanvas	sgcPDF_Renderer_FMX	El rasterizador por software con salida de mapa de bits FireMonkey y texto del sistema FMX. Delphi XE5 o posterior.

Normalmente no se utiliza el intérprete directamente. Cada backend tiene una clase renderizadora ya preparada que posee un intérprete y un canvas:

Renderizador	Salida	Plataformas
TsgcPDFRendererRaster	Archivo o stream PNG, archivo BMP, píxeles BGRA	Windows, Linux, macOS, iOS, Android
TsgcPDFRendererGDI	Contexto de dispositivo de Windows, archivo BMP	Windows
TsgcPDFRendererFMX	FireMonkey TBitmap	Todas las plataformas FMX

Propiedades comunes

Todas las clases renderizadoras exponen el intérprete como `Renderer` y el backend como `Canvas`, además de estas propiedades:

Propiedad	Descripción
RenderAnnotations: Boolean	Dibuja las anotaciones de página y los campos de formulario después del contenido de la página. Predeterminado True.
AnnotationMode: TsgcPDFAnnotRenderMode	**armView** (predeterminado) omite las anotaciones con el indicador NoView, **armPrint** dibuja solo las anotaciones con el indicador Print. Utilice armPrint para la impresión y para imágenes que deban coincidir con una copia impresa. No disponible en TsgcPDFRendererFMX, que siempre utiliza armView.

Escribir su propio backend

Para dibujar en otro destino (una biblioteca gráfica, un conversor de PDF a SVG, un grabador de pruebas), derive de **TsgcPDFRenderCanvas**, implemente sus métodos abstractos y páselo a `TsgcPDFRenderer.Create(ACanvas)`. Luego llame a `RenderPage(AParser, APageDict, AWidth, AHeight)`. El intérprete llama al canvas en el espacio de dispositivo: las coordenadas son píxeles con el eje Y creciendo hacia abajo, los colores son bytes RGB y el alfa va de 0 a 255. Los búferes de ráster son BGRA de 32 bits, de arriba hacia abajo y no premultiplicados.

Vea también

- TsgcPDFRendererRaster
- TsgcPDFRendererGDI
- Características de renderizado admitidas
- Renderizado y exportación

TsgcPDFRendererRaster

TsgcPDFRendererRaster (unidad sgcPDF_Renderer_Raster) renderiza páginas PDF a imágenes sin GDI, VCL ni FMX. Está escrito en Pascal puro, de modo que el mismo código se ejecuta en un servicio de Windows, un servidor web Linux o una herramienta de consola en macOS. Utilícelo para crear vistas previas y miniaturas de páginas en un servidor, para convertir PDF a PNG, o para alimentar píxeles de página a código de OCR o de procesamiento de imágenes.

Métodos

Método	Descripción
RenderPageToPNG(AParser; APageDict; AWidth, AHeight: Integer; const AFileName: string)	Renderiza una página a AWidth x AHeight píxeles sobre un fondo blanco y la guarda como PNG.
RenderPageToBitmap(AParser; APageDict; AWidth, AHeight: Integer; const AFileName: string)	Igual, guardado como un archivo BMP de 32 bits.
RenderPageToImage(AParser; APageDict; AWidth, AHeight: Integer)	Renderiza únicamente en el canvas. Lea los píxeles desde `Canvas.Bits` o guárdelos con los métodos del canvas.
RenderPage(AParser; APageDict; ARenderWidth, ARenderHeight: Integer; AOffsetX: Integer = 0; AOffsetY: Integer = 0)	Dibuja sobre el contenido actual del canvas sin redimensionarlo ni borrarlo. Ajuste primero el tamaño del canvas con `Canvas.SetSize` .
Render(AContentData: TBytes; AMediaBox: TsgcPDFRect; ...)	Renderiza un stream de contenido en bruto (sin recursos).

AParser es `TsgcPDFDocument.Parser` de un documento cargado y APageDict es `TsgcPDFDocument.Pages[i].PageDict`. Se aplica la rotación de página (/Rotate).

TsgcPDFRasterCanvas

La propiedad `Canvas` contiene el ráster de la página:

Miembro	Descripción
Width, Height	Tamaño del ráster en píxeles.
Bits: TBytes	Los píxeles: BGRA, de arriba hacia abajo, 4 * Width bytes por fila. El canal alfa siempre es 255.
SetSize(AWidth, AHeight)	Redimensiona el ráster y lo rellena de blanco.
Clear(R, G, B: Byte)	Rellena el ráster con un color.
SaveToPNGFile, SaveToPNGStream, SaveToBMPFile	Guarda el ráster.

Ejemplo: página a PNG

```
procedure RenderPageToPNG(const APDFFile, APNGFile: string; ADPI: Integer);
var
  oPDF: TsgcPDFDocument;
  oRenderer: TsgcPDFRendererRaster;
  oPage: TsgcPDFPage;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    oPage := oPDF.Pages[0];
    oRenderer := TsgcPDFRendererRaster.Create;
    try
      // page size in points (1/72 inch) to pixels
      oRenderer.RenderPageToPNG(oPDF.Parser, oPage.PageDict,
        Round(oPage.Width * ADPI / 72), Round(oPage.Height * ADPI / 72),
        APNGFile);
    finally
      oRenderer.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Ejemplo: todas las páginas, píxeles y streams

```
procedure RenderAllPagesToPixels(const APDFFile: string);
var
  oPDF: TsgcPDFDocument;
  oRenderer: TsgcPDFRendererRaster;
  oStream: TFileStream;
  i: Integer;
begin
  // extra folder with TrueType fonts for text in non embedded fonts
  sgcPDFRasterAddFontDirectory('/opt/myapp/fonts');
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    oRenderer := TsgcPDFRendererRaster.Create;
    try
      // annotations are drawn by default; armPrint draws the printable ones
      oRenderer.AnnotationMode := armPrint;
      for i := 0 to oPDF.PageCount - 1 do
      begin
        oRenderer.RenderPageToImage(oPDF.Parser, oPDF.Pages[i].PageDict,
          800, 1131);
        // Canvas.Bits: BGRA pixels, top-down, 4 * Width bytes per row
        WriteLn(Format('page %d: %d x %d, %d bytes', [i + 1,
          oRenderer.Canvas.Width, oRenderer.Canvas.Height,
          Length(oRenderer.Canvas.Bits)]));
        oStream := TFileStream.Create(Format('out\page%d.png', [i + 1]),
          fmCreate);
        try
          oRenderer.Canvas.SaveToPNGStream(oStream);
        finally
          oStream.Free;
        end;
      end;
    finally
      oRenderer.Free;
    end;
  finally
    oPDF.Free;
```

```
  end;
end;
```

Fuentes para texto no incrustado

El texto con fuentes incrustadas (TrueType, OpenType, CFF, Type 1 y Type 3) se dibuja a partir del programa de fuente del PDF. El texto con fuentes que no están incrustadas, como las 14 fuentes estándar, se dibuja con una fuente TrueType del sistema del mismo nombre o un sustituto cercano. El renderizador examina estas carpetas:

- Windows: %WINDIR%\Fonts y %LOCALAPPDATA%\Microsoft\Windows\Fonts.
- Linux: /usr/share/fonts, /usr/local/share/fonts, ~/.fonts y ~/.local/share/fonts.
- macOS: /System/Library/Fonts, /Library/Fonts y ~/Library/Fonts.
- Android: /system/fonts.

Llame a `sgcPDFRasterAddFontDirectory(const ADirectory: string)` antes de renderizar para agregar otra carpeta (se examina de forma recursiva). En un servidor Linux mínimo o un contenedor sin fuentes, instale un paquete de fuentes (por ejemplo fonts-dejavu o fonts-liberation) o distribuya sus propias fuentes y registre su carpeta.

Límites

El tamaño del ráster está limitado a CS_CANVAS_MAX_RASTER píxeles (67 108 864, por ejemplo 8192 x 8192) para proteger el proceso de asignaciones enormes. Divida las salidas más grandes en mosaicos con `RenderPage` y los parámetros de desplazamiento.

Vea también

- Arquitectura del renderizador
- Características de renderizado admitidas
- Linux
- TsgcPDFRendererGDI

TsgcPDFRendererGDI

TsgcPDFRendererGDI (unidad sgcPDF_Renderer_GDI) renderiza páginas PDF en un contexto de dispositivo de Windows. Utiliza el mismo intérprete de contenido que el renderizador ráster y dibuja mediante **TsgcPDFGDICanvas**. TsgcPDFViewer, TsgcPDFViewerControl, TsgcPDFThumbnailView, TsgcPDFExporter y la impresión se construyen sobre él. Solo Windows.

Métodos

Método	Descripción
RenderPage(ADC: HDC; AParser; APageDict; ARenderWidth, ARenderHeight: Integer; AOffsetX: Integer = 0; AOffsetY: Integer = 0)	Dibuja una página en un contexto de dispositivo: un canvas de formulario o control (`Canvas.Handle`), un mapa de bits en memoria o una impresora.
Render(ADC: HDC; AContentData: TBytes; AMediaBox: TsgcPDFRect; ...)	Dibuja un stream de contenido en bruto.
RenderPageToBitmap(AParser; APageDict; AWidth, AHeight: Integer; const AFileName: string)	Renderiza una página a un archivo BMP.
RenderToBitmap(AContentData: TBytes; AMediaBox: TsgcPDFRect; AWidth, AHeight: Integer; const AFileName: string)	Renderiza un stream de contenido en bruto a un archivo BMP.

Propiedades: `Renderer`, `Canvas`, `RenderAnnotations` y `AnnotationMode`, vea Arquitectura del renderizador.

Ejemplo: página a archivo de mapa de bits

```
procedure RenderPageWithGDI(const APDFFile, ABMPFile: string);
var
  oPDF: TsgcPDFDocument;
  oRenderer: TsgcPDFRendererGDI;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    oRenderer := TsgcPDFRendererGDI.Create;
    try
      // skip annotations and form fields, draw the page content only
      oRenderer.RenderAnnotations := False;
      oRenderer.RenderPageToBitmap(oPDF.Parser, oPDF.Pages[0].PageDict,
        1240, 1754, ABMPFile);
    finally
      oRenderer.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Ejemplo: pintar en un contexto de dispositivo

```
procedure PaintPage(ADC: HDC; APDF: TsgcPDFDocument; APageIndex, AWidth,
  AHeight: Integer);
var
  oRenderer: TsgcPDFRendererGDI;
begin
  oRenderer := TsgcPDFRendererGDI.Create;
  try
    // draws on any device context: a form canvas, a bitmap or a printer
    oRenderer.RenderPage(ADC, APDF.Parser, APDF.Pages[APageIndex].PageDict,
      AWidth, AHeight);
  finally
    oRenderer.Free;
  end;
end;
```

Vea también

- Arquitectura del renderizador
- TsgcPDFRendererRaster
- TsgcPDFViewerControl

Características de renderizado admitidas

Esta página enumera lo que dibuja el intérprete de contenido. Se aplica a todos los renderizadores (ráster, GDI y FMX) y a los visores construidos sobre ellos.

Gráficos

- Trayectos: líneas, curvas Bezier, rectángulos, relleno con las reglas nonzero y even-odd, trazo, recorte.
- Ancho de línea, cap, join, límite de miter y patrones de guiones.
- Operadores de color para todos los espacios de color (cs, CS, sc, SC, scn, SCN, g, rg, k y sus variantes de trazo).
- Alfa constante de relleno y trazo (ExtGState /ca y /CA).
- Modos de fusión: Normal y Multiply se componen exactamente. Los demás modos de fusión se dibujan como Normal.
- Degradados: basados en función (tipo 1), axial (tipo 2), radial (tipo 3), mallas Gouraud de forma libre y de forma reticular (tipos 4 y 5), parches Coons y de producto tensorial (tipos 6 y 7), utilizados con el operador sh y como patrones de degradado.
- Patrones de mosaico, coloreados y sin color.
- XObjects de formulario con su matriz y cuadro delimitador.

Espacios de color

DeviceGray, DeviceRGB, DeviceCMYK, CalGray, CalRGB, Lab, ICCBased (mediante su alternativa o número de componentes), Indexed, Separation y DeviceN, con funciones de transformación de tinta muestreadas (tipo 0), exponenciales (tipo 2) y de calculadora PostScript (tipo 4). CMYK se convierte a RGB con una fórmula simple, sin gestión de color ICC.

Imágenes

- XObjects de imagen e imágenes en línea (BI, ID, EI).
- Filtros: DCTDecode (JPEG baseline y progresivo), JPXDecode (JPEG 2000), CCITTFaxDecode (Group 3 y Group 4), JBIG2Decode, FlateDecode y LZWDecode con predictores, RunLengthDecode, ASCII85Decode y ASCIIHexDecode. Vea Decodificadores de imagen.
- 1, 2, 4, 8 y 16 bits por componente, arrays /Decode, máscaras de imagen (stencil masks), máscaras de clave de color, streams /Mask y máscaras suaves /SMask con /Matte.
- Colocación con rotación, inclinación y volteos, con reducción de escala suave.

Texto y fuentes

- Fuentes TrueType incrustadas (FontFile2), fuentes CFF y OpenType CFF (FontFile3), fuentes Type 1 (FontFile), y fuentes Type 3 (glifos dibujados como streams de contenido).
- Fuentes simples con codificaciones WinAnsi, MacRoman, Standard y /Differences, fuentes simbólicas, y fuentes compuestas (Type0) con Identity-H, Identity-V y los CMaps CJK predefinidos.
- Escritura vertical (Identity-V y los CMaps verticales, con métricas /W2).
- Espaciado de caracteres y palabras, escalado horizontal, elevación (rise), y los modos de renderizado de texto (fill, stroke, clip, invisible).
- Las fuentes que no están incrustadas se dibujan con una fuente instalada del sistema del mismo nombre o un sustituto. El texto pequeño se sobremuestrea para que siga siendo legible a resoluciones bajas.

Anotaciones y formularios

- Streams de apariencia de anotación (/AP /N), con el estado /AS de las casillas de verificación y los botones de opción, los indicadores Hidden, NoView y Print, y el mapeo del cuadro delimitador de la apariencia al / Rect de la anotación.

- Un dibujo de reserva simple para los tipos de anotación comunes sin stream de apariencia.
- Cuando el formulario establece **/NeedAppearances** en true, los campos de texto, los campos de selección y los botones sin apariencia normal se dibujan a partir de sus valores de campo. `sgcPDFWidgetAppearanceGenerated(AParser, AWidget)` indica si un widget se dibuja de esta manera.
- Establezca `RenderAnnotations := False` para dibujar solo el contenido de la página, y `AnnotationMode := arm-Print` para dibujar lo que imprimiría una impresora.

No admitido

- Los diccionarios de máscara suave en ExtGState (/SMask distinto de /None) se ignoran, y los grupos de transparencia no se aíslan ni se recortan (knockout).
- El contenido opcional (capas) no se evalúa: se dibuja el contenido de las capas ocultas.
- Los modos de fusión distintos de Normal y Multiply se aproximan como Normal.

Vea también

- Arquitectura del renderizador
- Decodificadores de imagen
- Fuentes OpenType

TsgcPDFViewerControl

TsgcPDFViewerControl (unidad sgcPDF_ViewerVCL) es un control VCL que muestra documentos PDF dentro de sus formularios. Arrástrelo a un formulario, o créelo en tiempo de ejecución, y cargue un archivo. Las páginas se renderizan mediante TsgcPDFRendererGDI en segundo plano en una caché de páginas acotada, de modo que los documentos grandes se abren rápidamente y se desplazan con suavidad. Está en la página de paleta **sgcPDF** junto con los controles complementarios TsgcPDFThumbnailView, TsgcPDFOutlineView y TsgcPDFCommentsView. Características: diseño continuo o de página única, modos de zoom, navegación con teclado, rueda del ratón y herramienta de mano, selección y copia de texto, búsqueda con resaltado, navegación por enlaces, relleno interactivo de formularios, revisión y edición de anotaciones e impresión. Compatible con alta densidad de píxeles (High DPI) en Delphi 10.3 y posterior.

Propiedades publicadas

Propiedad	Descripción
Document: TsgcPDFDocument	Muestra un TsgcPDFDocument del que usted es propietario. Déjelo vacío para que el control cargue los archivos por sí mismo.
FileName: string	Carga un archivo (también en tiempo de diseño).
PageIndex: Integer	Página actual, con base cero.
Zoom: Double	Zoom en porcentaje. Establecerlo cambia ZoomMode a zmCustom.
ZoomMode: TsgcPDFViewerZoomMode	zmFitWidth (predeterminado), zmFitPage, zmActualSize o zmCustom.
ViewMode: TsgcPDFViewerViewMode	pvmContinuous (predeterminado, todas las páginas una debajo de otra) o pvmSinglePage.
PageGap: Integer	Espacio entre páginas en píxeles. Predeterminado 10.
BackgroundColor: TColor	Color alrededor de las páginas. Predeterminado clAppWorkSpace.
ShowPageShadow: Boolean	Dibuja una sombra bajo cada página. Predeterminado True.
CacheSize: Integer	Número de páginas renderizadas conservadas en memoria. Predeterminado 12.
SelectionMode: TsgcPDFViewerSelectionMode	psmText (predeterminado) selecciona texto con el ratón, psmHand arrastra las páginas.
SelectionColor, SearchColor, SearchCurrentColor: TColor	Colores de la selección de texto, de todas las coincidencias de búsqueda y de la coincidencia actual.
FormFilling: Boolean	Permite al usuario rellenar campos de texto, casillas de verificación, botones de opción, listas y listas desplegables. Predeterminado True.
OpenLinksInBrowser: Boolean	Abre los enlaces http, https y mailto con la aplicación predeterminada. Predeterminado False.
AnnotationEditing: Boolean	Permite al usuario seleccionar, mover, redimensionar y eliminar anotaciones, editar su texto y añadir otras nuevas. Los enlaces se seleccionan en lugar de seguirse. Si se establece en False, borra la selección y conserva la herramienta, porque

Propiedad	Descripción
	las herramientas también funcionan con AnnotationEditing en False. Valor predeterminado False.
AnnotationTool: TsgcPDFViewerAnnotationTool	Herramienta para la siguiente anotación colocada en una página, con AnnotationEditing en True o False: vatNone (predeterminado), vatNote, vatFreeText, vatHighlight, vatUnderline, vatStrikeOut, vatLine, vatArrow, vatSquare, vatCircle, vatInk o vatStamp.
AnnotationAuthor: string	Autor de las anotaciones y respuestas que añade el usuario.
AnnotationColor: TColor	Color de las anotaciones nuevas. clDefault (predeterminado) usa el color habitual de cada herramienta.
AnnotationStampName: string	Nombre del sello colocado por vatStamp, como Approved (predeterminado), Draft o Confidential.
ShowAnnotationPopups: Boolean	Abre un popup al hacer clic en una anotación con contenido o respuestas. Valor predeterminado True.

Eventos

Evento	Descripción
OnPageChange: TNotifyEvent	La página actual cambió.
OnZoomChange: TNotifyEvent	El zoom cambió.
OnLinkClick(Sender; APageIndex: Integer; const AURI: string; var AHandled: Boolean)	Se hizo clic en un enlace. AURI está vacío para un enlace a una página del documento. Establezca AHandled en True para cancelar la acción predeterminada.
OnFieldChange(Sender; AField: TsgcPDFFormField)	El usuario cambió el valor de un campo de formulario.
OnSearchProgress(Sender; APageIndex, APageCount: Integer; var ACancel: Boolean)	Se llama por cada página mientras FindText busca. Establezca ACancel para detener.
OnAnnotationClick(Sender; AAnnotation: TsgcPDFAnnotation; var AHandled: Boolean)	Se hizo clic en una anotación con contenido o respuestas. Establezca AHandled en True para omitir el popup.
OnAnnotationSelect(Sender; AAnnotation: TsgcPDFAnnotation)	La anotación seleccionada ha cambiado. AAnnotation es nil cuando se borró la selección.
OnAnnotationAdd(Sender; AAnnotation: TsgcPDFAnnotation)	El usuario añadió una anotación con una herramienta, o una respuesta en el popup.
OnAnnotationChange(Sender; AAnnotation: TsgcPDFAnnotation)	El usuario movió, redimensionó o editó una anotación.
OnAnnotationDelete(Sender; AAnnotation: TsgcPDFAnnotation; var AAllow: Boolean)	Una anotación está a punto de eliminarse junto con sus respuestas. Establezca AAllow en False para conservarla.

Métodos públicos

Método	Descripción
LoadFromFile, LoadFromStream, Close	Abre y cierra documentos.
GotoPage, NextPage, PrevPage, FirstPage, LastPage	Navegación.
ZoomIn, ZoomOut	Cambia el zoom un paso.
Print(AFromPage: Integer = 0; AToPage: Integer = -1; const ATitle: string = '')	Imprime un rango de páginas (con base cero, -1 es la última página) en la impresora actual, con las anotaciones en modo de impresión.
RenderPageToBitmap(APageIndex: Integer; ABitmap: TBitmap; AZoomPercent: Double = 100; ADPI: Integer = 0)	Renderiza una página en un mapa de bits VCL.
FindText(const AText: WideString; ACaseSensitive: Boolean = False; AWholeWord: Boolean = False; ABackward: Boolean = False): Boolean	Busca y resalta todas las coincidencias del documento y se desplaza hasta la primera.
FindNext, FindPrevious, ClearSearch, GetSearchMatch(AIndex)	Se desplaza entre coincidencias. SearchResults es el número de coincidencias y SearchIndex la actual.
SelectText(APageIndex, AFirstChar, ACharCount), SelectPage, SelectAll, ClearSelection, CopyToClipboard	Selección de texto. SelectedText y HasSelection leen la selección.
PageText(APageIndex): TsgcPDFTextExtractor	Texto de una página con posiciones de caracteres (propiedad del visor).
DeviceToPage, PageToDevice, GetPageRect, PageRotation	Convierte entre píxeles de cliente y el espacio de usuario de PDF.
WidgetCount, GetWidget(AIndex), EditWidget	Accede a los widgets de campo de formulario y comienza a editar uno.
ClearCache, RenderVisiblePages, CachedPageCount	Control de la caché de páginas.
AnnotationAt(X, Y: Integer): TsgcPDFAnnotation	La anotación superior en un punto del cliente, nil cuando no hay ninguna. Se omiten las respuestas, los popups, los widgets de formulario y las anotaciones ocultas, y los enlaces solo se devuelven con AnnotationEditing en True.
SelectedAnnotation, SelectAnnotation(AAnnotation), DeleteSelectedAnnotation: Boolean	La anotación seleccionada. SelectAnnotation(nil) borra la selección. DeleteSelectedAnnotation funciona con AnnotationEditing en True, dispara OnAnnotationDelete y devuelve True cuando se eliminó la anotación.
ShowAnnotationPopup(AAnnotation), CloseAnnotationPopup	Abre el popup de una anotación desde código, y lo cierra.
GotoAnnotation(AAnnotation)	Desplaza la anotación a la vista. Con AnnotationEditing en True también se selecciona.
AnnotationsChanged	Llámelo después de modificar Document.Annotations en código. El visor se re-

Método	Descripción
	pinta y descarta la selección o el popup de una anotación eliminada.

Ratón y teclado

- La rueda del ratón desplaza, Ctrl + rueda hace zoom.
- Las teclas de flecha, Av Pág, Re Pág, Espacio, Inicio y Fin desplazan y cambian de página.
- Ctrl+A selecciona todo el texto del documento, Ctrl+C y Ctrl+Insert copian la selección.
- Tab se desplaza entre los campos de formulario durante el relleno.
- Un clic en una anotación con contenido o respuestas abre su popup. Con AnnotationEditing en True, un clic selecciona una anotación, arrastrar la mueve, los tiradores la redimensionan, Supr la elimina y las teclas de flecha la mueven (10 puntos con Mayús).
- Esc durante el arrastre de una herramienta, de un movimiento o de un cambio de tamaño cancela solo ese arrastre. En otro caso cierra el popup de la anotación, borra la selección y cancela la herramienta de anotación.

Ejemplo: visor con miniaturas y marcadores

```
constructor TMainForm.CreateViewer(AOwner: TComponent);
begin
  CreateNew(AOwner);
  Width := 1100;
  Height := 800;
  // page thumbnails on the left
  FThumbnails := TsgcPDFThumbnailView.Create(Self);
  FThumbnails.Parent := Self;
  FThumbnails.Align := alLeft;
  FThumbnails.Width := 160;
  FThumbnails.ThumbnailWidth := 120;
  // bookmarks (outline) tree on the right
  FOutline := TsgcPDFOutlineView.Create(Self);
  FOutline.Parent := Self;
  FOutline.Align := alRight;
  FOutline.Width := 220;
  // the viewer fills the rest of the form
  FViewer := TsgcPDFViewerControl.Create(Self);
  FViewer.Parent := Self;
  FViewer.Align := alClient;
  FViewer.ZoomMode := zmFitWidth;
  FViewer.ViewMode := pvmContinuous;
  FViewer.OnPageChange := OnPageChangeEvent;
  FViewer.OnLinkClick := OnLinkClickEvent;
  FViewer.OnFieldChange := OnFieldChangeEvent;
  FViewer.OnSearchProgress := OnSearchProgressEvent;
  // link the side panels to the viewer
  FThumbnails.Viewer := FViewer;
  FOutline.Viewer := FViewer;
  FViewer.LoadFromFile('sample.pdf');
end;
```

Ejemplo: eventos

```
procedure TMainForm.OnPageChangeEvent(Sender: TObject);
begin
  Caption := Format('Page %d of %d, zoom %.0f%%',
    [FViewer.PageIndex + 1, FViewer.PageCount, FViewer.Zoom]);
end;

procedure TMainForm.OnLinkClickEvent(Sender: TObject; APageIndex: Integer;
  const AURI: string; var AHandled: Boolean);
```

```
begin
  // AURI is empty for links to a page of the same document
  if AURI <> '' then
    AHandled := MessageDlg('Open ' + AURI + '?', mtConfirmation,
      [mbYes, mbNo], 0) <> mrYes;
end;

procedure TMainForm.OnFieldChangeEvent(Sender: TObject;
  AField: TsgcPDFFormField);
begin
  // a form field was edited in the viewer: save the filled form
  FViewer.Document.SaveToFile('filled.pdf');
end;

procedure TMainForm.OnSearchProgressEvent(Sender: TObject;
  APageIndex, APageCount: Integer; var ACancel: Boolean);
begin
  Caption := Format('Searching page %d of %d', [APageIndex + 1, APageCount]);
end;
```

Ejemplo: búsqueda y copia

```
procedure TMainForm.SearchAndCopy;
begin
  // every match of the document is highlighted, the first one is shown
  if FViewer.FindText('invoice', False, True) then
  begin
    ShowMessage(Format('%d matches', [FViewer.SearchResults]));
    FViewer.FindNext;
  end;
  // select the text of the current page and copy it to the clipboard
  FViewer.SelectPage(FViewer.PageIndex);
  if FViewer.HasSelection then
    FViewer.CopyToClipboard;
end;
```

Ejemplo: impresión

```
procedure TMainForm.PrintPages;
begin
  // pages 1 to 3 (zero based indexes) on the default printer
  FViewer.Print(0, 2, 'My document');
end;
```

Anotaciones

El visor dibuja las anotaciones de `Document.Annotations`. Al hacer clic en una anotación que tiene contenido o respuestas se abre un popup con su autor, fecha, contenido y respuestas. Gestione OnAnnotationClick, o establezca ShowAnnotationPopups en False, para mostrar su propia interfaz.
Con **AnnotationEditing** en True el usuario también puede modificar las anotaciones:

- Un clic selecciona una anotación y muestra sus tiradores. Arrastre la anotación para moverla, o arrastre un tirador para redimensionarla.
- Supr elimina la anotación seleccionada con sus respuestas. OnAnnotationDelete puede rechazarlo.
- Las teclas de flecha mueven la anotación seleccionada 1 punto, o 10 puntos con Mayús.
- El texto del popup se puede editar, y su cuadro Reply añade una respuesta escrita por AnnotationAuthor.
- Esc durante el arrastre de una herramienta, de un movimiento o de un cambio de tamaño cancela solo ese arrastre. En otro caso cierra el popup, borra la selección y cancela la herramienta actual.

Establezca **AnnotationTool** para colocar una anotación nueva: haga clic en una página, o arrastre sobre ella, y la anotación se crea con AnnotationAuthor y AnnotationColor. Se dispara OnAnnotationAdd y la herramienta vuelve a vatNone.

Herramienta	Anotación colocada
vatNone	Ninguna herramienta (predeterminado). Los clics seleccionan anotaciones.
vatNote	Nota adhesiva (TsgcPDFTextAnnotation).
vatFreeText	Cuadro de texto (TsgcPDFFreeTextAnnotation).
vatHighlight, vatUnderline, vatStrikeOut	Marca de texto (TsgcPDFHighlightAnnotation) sobre el texto que se selecciona con el ratón.
vatLine, vatArrow	Línea, o línea con una flecha en el extremo (TsgcPDFLineAnnotation).
vatSquare, vatCircle	Rectángulo o elipse (TsgcPDFShapeAnnotation).
vatInk	Dibujo a mano alzada (TsgcPDFInkAnnotation).
vatStamp	Sello con el nombre AnnotationStampName (TsgcPDFStampAnnotation).

Guardar sigue siendo tarea de la aplicación. El visor solo modifica el documento en memoria, y `Document.Annotations.Modified` indica si hay algo que guardar. Llame a `Document.SaveIncremental` para añadir los cambios y mantener válidas las firmas existentes, o a `Document.SaveToFile` para un guardado completo. Vea Anotaciones. Después de modificar `Document.Annotations` en código, llame a AnnotationsChanged. Para mostrar las anotaciones como una lista de comentarios junto al visor, enlace con él un TsgcPDFCommentsView.

Ejemplo: anotaciones

```
procedure TMainForm.SetupAnnotations;
begin
  FViewer.AnnotationEditing := True;
  FViewer.AnnotationAuthor := 'Reviewer';
  // clDefault (the default) uses the usual colour of each tool
  FViewer.AnnotationColor := clRed;
  FViewer.AnnotationStampName := 'Draft';
  FViewer.OnAnnotationSelect := OnAnnotationSelectEvent;
  FViewer.OnAnnotationAdd := OnAnnotationAddEvent;
  FViewer.OnAnnotationChange := OnAnnotationChangeEvent;
  FViewer.OnAnnotationDelete := OnAnnotationDeleteEvent;
  FViewer.OnAnnotationClick := OnAnnotationClickEvent;
end;

procedure TMainForm.NoteButtonClick(Sender: TObject);
begin
  // the next click on a page places a sticky note
  FViewer.AnnotationTool := vatNote;
end;

procedure TMainForm.DeleteButtonClick(Sender: TObject);
begin
  FViewer.DeleteSelectedAnnotation;
end;

procedure TMainForm.SaveButtonClick(Sender: TObject);
begin
  if FViewer.Document.Annotations.Modified then
    // the original bytes are kept, and so are the signatures of the file
    FViewer.Document.SaveIncremental(ChangeFileExt(FViewer.FileName, '') +
      '_reviewed.pdf');
end;

procedure TMainForm.OnAnnotationSelectEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation);
begin
```

```
  // AAnnotation is nil when the selection is cleared
  DeleteButton.Enabled := AAnnotation <> nil;
end;

procedure TMainForm.OnAnnotationAddEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation);
begin
  StatusBar.SimpleText := Format('%s added on page %d',
    [AAnnotation.ClassName, AAnnotation.PageIndex + 1]);
end;

procedure TMainForm.OnAnnotationChangeEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation);
begin
  StatusBar.SimpleText := 'Changed: ' + AAnnotation.Contents;
end;

procedure TMainForm.OnAnnotationDeleteEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation; var AAllow: Boolean);
begin
  AAllow := MessageDlg('Delete this annotation and its replies?',
    mtConfirmation, [mbYes, mbNo], 0) = mrYes;
end;

procedure TMainForm.OnAnnotationClickEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation; var AHandled: Boolean);
begin
  // no popup for stamps in this application
  AHandled := AAnnotation is TsgcPDFStampAnnotation;
end;
```

Relleno de formularios

Con FormFilling en True, al hacer clic en un campo de texto o de selección se abre un editor sobre el campo, y al hacer clic en una casilla de verificación o un botón de opción se alterna su estado. Los valores de los campos se escriben en el documento (TsgcPDFDocument.AcroForm) y la página se vuelve a pintar. Gestione OnFieldChange para reaccionar, y guarde el documento con `Document.SaveToFile` para conservar los valores rellenados.

Vea también

- TsgcPDFThumbnailView
- TsgcPDFOutlineView
- TsgcPDFCommentsView
- TsgcPDFViewerFMX
- TsgcPDFViewer
- Anotaciones

TsgcPDFThumbnailView

TsgcPDFThumbnailView (unidad sgcPDF_ViewerVCL) muestra una columna desplazable de miniaturas de página para un TsgcPDFViewerControl. Haga clic en una miniatura para ir a esa página. La página actual se resalta, y las miniaturas se renderizan bajo demanda y se almacenan en caché.

Miembro	Descripción
Viewer: TsgcPDFViewerControl	El visor cuyo documento se muestra. Las miniaturas siguen su documento y su página actual.
ThumbnailWidth: Integer	Ancho de una miniatura en píxeles. Predeterminado 120.
ItemCount	Número de miniaturas (páginas).
GetItemRect(AIndex): TRect, ItemAtPos(X, Y): Integer	Disposición de las miniaturas en coordenadas de cliente.
RenderVisibleThumbnails, CachedThumbnailCount	Renderizado y control de caché.

Se publican las propiedades VCL habituales (Align, Color, Font, PopupMenu...) y OnClick, OnMouseDown y OnResize. Vea el ejemplo en TsgcPDFViewerControl.

Vea también

- TsgcPDFViewerControl
- TsgcPDFOutlineView

TsgcPDFOutlineView

TsgcPDFOutlineView (unidad sgcPDF_ViewerVCL) es una vista de árbol que muestra los marcadores (esquemas) del documento de un TsgcPDFViewerControl. Al seleccionar un nodo, el visor se desplaza hasta el destino del marcador.

Miembro	Descripción
Viewer: TsgcPDFViewerControl	El visor que se sigue. El árbol se reconstruye cuando el visor carga otro documento.
RebuildOutline	Vuelve a cargar el árbol a partir del documento.
NavigateToNode(ANode: TTreeNode)	Va al destino de un nodo.

Desciende de TCustomTreeView y publica las propiedades habituales de una vista de árbol (Items, Indent, ShowButtons, ShowLines, ShowRoot, HideSelection...) y los eventos OnClick y OnChange.

Vea también

- TsgcPDFViewerControl
- TsgcPDFCommentsView
- Marcadores

TsgcPDFCommentsView

TsgcPDFCommentsView (unidad sgcPDF_ViewerVCL) es una vista de árbol que enumera los comentarios del documento de un TsgcPDFViewerControl, como el panel de comentarios de un lector de PDF. Se enumeran todas las anotaciones excepto los campos de formulario, los enlaces y los popups.

Miembro	Descripción
Viewer: TsgcPDF-ViewerControl	El visor que se sigue. La lista se reconstruye cuando el visor carga o cierra un documento y cuando cambian sus anotaciones.
GroupByPage: Boolean	Enumera los comentarios bajo un nodo por página, con el texto Page N (count). Cuando es False, los comentarios forman una lista plana en el orden de las páginas. Valor predeterminado True.
ShowReplies: Boolean	Muestra las respuestas de un comentario como sus nodos hijos. Cuando es False, solo se enumeran los comentarios de nivel superior. Valor predeterminado True.
RebuildComments	Vuelve a cargar la lista a partir del documento del visor. La lista ya sigue al visor, y después de modificar Document.Annotations en código el método AnnotationsChanged del visor también la reconstruye.
SelectedAnnotation: TsgcPDFAnnotation	Solo lectura. La anotación del nodo seleccionado, nil cuando no hay ningún nodo seleccionado o es un nodo de página.

Cada nodo muestra `Author, date: contents`. El contenido se muestra en una línea, y los textos largos se cortan a 120 caracteres. Cuando el contenido está vacío, se muestra el tipo de la anotación, como Highlight o Stamp (Approved). Una anotación sin autor muestra (no author), y la fecha se omite cuando la anotación no tiene ninguna.

- Al seleccionar un nodo, el visor se desplaza hasta la anotación con GotoAnnotation. Con AnnotationEditing del visor en True, la anotación también se selecciona en el visor.
- Un doble clic abre el popup de la anotación.
- La tecla Supr elimina la anotación seleccionada y sus respuestas mientras AnnotationEditing del visor sea True. OnAnnotationDelete del visor puede rechazarlo.
- Cuando el usuario selecciona una anotación en el visor, su nodo se selecciona en la lista.

Desciende de TCustomTreeView y publica las propiedades habituales de una vista de árbol (Items, Indent, ShowButtons, ShowLines, ShowRoot, HideSelection...) y los eventos OnClick y OnChange. El árbol es de solo lectura, y HideSelection es False de forma predeterminada para que el comentario seleccionado siga visible.

Ejemplo: panel de comentarios

```
procedure TMainForm.CreateCommentsPanel;
begin
  FComments := TsgcPDFCommentsView.Create(Self);
  FComments.Parent := Self;
  FComments.Align := alRight;
  FComments.Width := 280;
  // one flat list instead of the "Page N (count)" groups
  FComments.GroupByPage := False;
  FComments.ShowReplies := True;
  FComments.OnChange := OnCommentChangeEvent;
  // from now on the list follows the document, the annotations and the
  // selection of the viewer
  FComments.Viewer := FViewer;
  // a selected comment is also selected in the viewer, Delete removes it
  FViewer.AnnotationEditing := True;
end;

procedure TMainForm.OnCommentChangeEvent(Sender: TObject; Node: TTreeNode);
```

```
var
  oAnnot: TsgcPDFAnnotation;
begin
  // nil for a page group node
  oAnnot := FComments.SelectedAnnotation;
  if oAnnot <> nil then
    StatusBar.SimpleText := Format('%s, page %d', [oAnnot.Author,
      oAnnot.PageIndex + 1]);
end;
```

Vea también

- TsgcPDFViewerControl
- TsgcPDFOutlineView
- TsgcPDFCommentsViewFMX
- Anotaciones

TsgcPDFViewerFMX

TsgcPDFViewerFMX (unidad sgcPDF_ViewerFMX) es un control FireMonkey que muestra documentos PDF en Windows, macOS, iOS, Android y Linux (FMXLinux). Las páginas se dibujan mediante el rasterizador por software en Pascal puro (TsgcPDFRendererRaster con salida de mapa de bits FMX) en un diseño vertical continuo, y se conservan en una caché de páginas acotada. Necesita Delphi XE5 o posterior y se instala con los paquetes FMX, en la página de paleta **sgcPDF**.

Propiedades publicadas

Propiedad	Descripción
Document: TsgcPDFDocument	Muestra un TsgcPDFDocument del que usted es propietario.
PageIndex: Integer	Página actual, con base cero.
Zoom: Single	Factor de zoom, 1 es el 100 por ciento. Establecerlo cambia ZoomMode a fzmCustom.
ZoomMode: TsgcPDFViewerFMXZoomMode	fzmFitWidth (predeterminado), fzmFitPage o fzmCustom.
PageGap: Single	Espacio entre páginas.
BackgroundColor: TAlphaColor	Color alrededor de las páginas. Predeterminado $FF808080.
CacheSize: Integer	Número de páginas renderizadas conservadas en memoria. Predeterminado 8.
OnPageChange, OnZoomChange, OnLayerChange: TNotifyEvent	Notificaciones de página, zoom y visibilidad de capas.
OnLinkClick(Sender; APageIndex: Integer; const AURI: string; var AHandled: Boolean)	Se tocó o hizo clic en un enlace. Los enlaces a páginas del documento se siguen a menos que se establezca AHandled.
AnnotationEditing: Boolean	Permite al usuario seleccionar, mover, redimensionar y eliminar anotaciones, editar su texto y añadir otras nuevas. Si se establece en False, borra la selección y conserva la herramienta, porque las herramientas también funcionan con AnnotationEditing en False. Un popup abierto se vuelve a abrir en el nuevo modo. Valor predeterminado False.
AnnotationTool: TsgcPDFViewerAnnotationTool	Herramienta para la siguiente anotación colocada en una página, con AnnotationEditing en True o False: vatNone (predeterminado), vatNote, vatFreeText, vatHighlight, vatUnderline, vatStrikeOut, vatLine, vatArrow, vatSquare, vatCircle, vatInk o vatStamp.
AnnotationAuthor: string	Autor de las anotaciones y respuestas que añade el usuario.
AnnotationColor: TAlphaColor	Color de las anotaciones nuevas. TAlphaColorRec.Null (predeterminado) usa el color habitual de cada herramienta.
AnnotationStampName: string	Nombre del sello colocado por vatStamp. Valor predeterminado Approved.

Propiedad	Descripción
ShowAnnotationPopups: Boolean	Abre un popup al tocar o hacer clic en una anotación con contenido o respuestas. Valor predeterminado True.
OnAnnotationClick(Sender; AAnnotation: TsgcPDFAnnotation; var AHandled: Boolean)	Se tocó o se hizo clic en una anotación con contenido o respuestas. Establezca AHandled en True para omitir el popup.
OnAnnotationSelect(Sender; AAnnotation: TsgcPDFAnnotation)	La anotación seleccionada ha cambiado. AAnnotation es nil cuando se borró la selección.
OnAnnotationAdd(Sender; AAnnotation: TsgcPDFAnnotation)	El usuario añadió una anotación con una herramienta, o una respuesta en el popup.
OnAnnotationChange(Sender; AAnnotation: TsgcPDFAnnotation)	El usuario movió, redimensionó o editó una anotación.
OnAnnotationDelete(Sender; AAnnotation: TsgcPDFAnnotation; var AAllow: Boolean)	Una anotación está a punto de eliminarse junto con sus respuestas. Establezca AAllow en False para conservarla.

Métodos públicos

LoadFromFile, LoadFromStream, Close, GotoPage, NextPage, PrevPage, FirstPage, LastPage, ZoomIn, ZoomOut, ClearCache, SetScrollPosition, GetPageRect, DeviceToPage, PageToDevice, PageRotation y CachedPageCount. Capas (contenido opcional): GetLayers, SetLayerVisible y LayerVisible, como en TsgcPDFViewerControl. Solo lectura: PageCount, ScrollX, ScrollY, ContentWidth, ContentHeight y FileName.
Anotaciones: AnnotationAt(X, Y: Single), SelectedAnnotation (lectura y escritura), SelectAnnotation, DeleteSelectedAnnotation (devuelve True cuando se eliminó la anotación), ShowAnnotationPopup, CloseAnnotationPopup, GotoAnnotation y AnnotationsChanged, que se llama después de modificar Document.Annotations en código. Funcionan como en TsgcPDFViewerControl: DeleteSelectedAnnotation solo funciona con AnnotationEditing en True, y GotoAnnotation desplaza la anotación a la vista y, con AnnotationEditing en True, también la selecciona.

Táctil, ratón y teclado

- Arrastre con el dedo o el ratón para desplazar y desplazar la vista.
- Pellizque para hacer zoom (el control activa el gesto interactivo Zoom).
- La rueda del ratón desplaza, Mayús + rueda desplaza hacia los lados, Ctrl + rueda hace zoom.
- Av Pág, Re Pág, Inicio, Fin y las teclas de flecha navegan.
- Con AnnotationEditing en False, toque un enlace para seguirlo, y toque una anotación con contenido o respuestas para abrir su popup.
- Con AnnotationEditing en True, un toque selecciona una anotación y un toque en un espacio vacío borra la selección. Un doble clic abre el popup. Arrastrar la anotación seleccionada la mueve, arrastrar uno de sus tiradores la redimensiona, y arrastrar en cualquier otro lugar desplaza la vista. Supr elimina la anotación seleccionada, y las teclas de flecha la mueven 1 punto, o 10 puntos con Mayús.
- Esc durante el arrastre de una herramienta, de un movimiento o de un cambio de tamaño cancela solo ese arrastre. En otro caso cierra el popup de la anotación, borra la selección y cancela la herramienta de anotación.

Ejemplo

```
constructor TMainForm.CreateViewer(AOwner: TComponent);
begin
  CreateNew(AOwner);
  FViewer := TsgcPDFViewerFMX.Create(Self);
  FViewer.Parent := Self;
  FViewer.Align := TAlignLayout.Client;
  FViewer.ZoomMode := fzmFitWidth;
  FViewer.BackgroundColor := TAlphaColors.Lightgray;
```

```
  FViewer.OnPageChange := OnPageChangeEvent;
  // on mobile, open a file from the application documents folder
  FViewer.LoadFromFile('sample.pdf');
end;

procedure TMainForm.OnPageChangeEvent(Sender: TObject);
begin
  Caption := Format('Page %d of %d', [FViewer.PageIndex + 1,
    FViewer.PageCount]);
end;
```

Anotaciones

El visor FMX tiene la misma compatibilidad con anotaciones que TsgcPDFViewerControl, con las mismas herramientas y eventos. Al tocar o hacer clic en una anotación que tiene contenido o respuestas se abre un popup con su autor, fecha, contenido y respuestas. Con AnnotationEditing en True el texto del popup se puede editar y su cuadro Reply añade una respuesta escrita por AnnotationAuthor.

- Con AnnotationEditing en True, un toque o clic selecciona una anotación y un doble clic abre su popup. Arrastrar la anotación seleccionada la mueve, sus tiradores la redimensionan y Supr la elimina con sus respuestas. Las teclas de flecha mueven la anotación seleccionada 1 punto, o 10 puntos con Mayús. Cada cambio dispara OnAnnotationChange.
- Esc durante el arrastre de una herramienta, de un movimiento o de un cambio de tamaño cancela solo ese arrastre. En otro caso cierra el popup, borra la selección y cancela la herramienta actual.
- AnnotationTool coloca una anotación con un toque o un arrastre y después vuelve a vatNone. Con AnnotationEditing en True, la anotación nueva queda seleccionada. Una nota nueva abre su popup, cuando ShowAnnotationPopups es True, para poder escribir su texto. Un texto libre nuevo abre un editor en el propio lugar. Ctrl+Intro, o un toque fuera del editor, añade la anotación con el texto escrito. Esc, o un texto vacío, la descarta.
- El visor FMX no tiene selección de texto, de modo que vatHighlight, vatUnderline y vatStrikeOut marcan el texto que queda dentro del rectángulo que se arrastra, un cuadrilátero por línea. Cuando el rectángulo no contiene texto, no se añade nada y la herramienta sigue activa.
- Las posiciones están en coordenadas del control (Single): AnnotationAt(X, Y) y PageToDevice(APageIndex, APDFX, APDFY): TPointF.

Guardar sigue siendo tarea de la aplicación, como en el visor VCL: llame a `Document.SaveIncremental` o a `Document.SaveToFile` cuando `Document.Annotations.Modified` sea True. Después de guardar, `Document.Annotations.ResetModified` borra ese indicador, y el siguiente guardado todavía escribe todos los cambios hechos desde que se cargó el documento. Vea Anotaciones. Para mostrar las anotaciones como una lista de comentarios junto al visor, enlace con él un TsgcPDFCommentsViewFMX.

Ejemplo: anotaciones

```
procedure TMainForm.SetupAnnotations;
begin
  FViewer.AnnotationEditing := True;
  FViewer.AnnotationAuthor := 'Reviewer';
  // TAlphaColorRec.Null (the default) uses the usual colour of each tool
  FViewer.AnnotationColor := TAlphaColorRec.Red;
  FViewer.OnAnnotationAdd := OnAnnotationAddEvent;
  FViewer.OnAnnotationChange := OnAnnotationChangeEvent;
  FViewer.OnAnnotationDelete := OnAnnotationDeleteEvent;
end;

procedure TMainForm.HighlightButtonClick(Sender: TObject);
begin
  // drag a rectangle over the text to highlight it
  FViewer.AnnotationTool := vatHighlight;
end;

procedure TMainForm.SaveButtonClick(Sender: TObject);
begin
  if FViewer.Document.Annotations.Modified then
    FViewer.Document.SaveIncremental(TPath.Combine(TPath.GetDocumentsPath,
      'reviewed.pdf'));
```

```
end;

procedure TMainForm.OnAnnotationAddEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation);
begin
  Caption := Format('%s added on page %d', [AAnnotation.ClassName,
    AAnnotation.PageIndex + 1]);
end;

procedure TMainForm.OnAnnotationChangeEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation);
begin
  Caption := 'Changed: ' + AAnnotation.Contents;
end;

procedure TMainForm.OnAnnotationDeleteEvent(Sender: TObject;
  AAnnotation: TsgcPDFAnnotation; var AAllow: Boolean);
begin
  // only the annotations of this user can be deleted
  AAllow := AAnnotation.Author = FViewer.AnnotationAuthor;
end;
```

Renderizado a un mapa de bits FMX

TsgcPDFRendererFMX (unidad sgcPDF_Renderer_FMX) renderiza una página en un TBitmap de FMX, por ejemplo para una vista previa personalizada o para compartir una imagen de página. El texto con fuentes que no están incrustadas y no están disponibles como archivos TrueType se dibuja con el motor de texto FMX de la plataforma.

```
procedure RenderPageToFMXBitmap(ADocument: TsgcPDFDocument;
  APageIndex: Integer; ABitmap: TBitmap);
var
  oRenderer: TsgcPDFRendererFMX;
begin
  oRenderer := TsgcPDFRendererFMX.Create;
  try
    // 800 x 1131 pixels, the bitmap is resized to the page raster
    oRenderer.RenderPageToBitmap(ADocument.Parser,
      ADocument.Pages[APageIndex].PageDict, 800, 1131, ABitmap);
  finally
    oRenderer.Free;
  end;
end;
```

La selección de texto, la búsqueda y el relleno de formularios solo están disponibles en el visor VCL.

Vea también

- TsgcPDFViewerControl
- TsgcPDFCommentsViewFMX
- macOS, iOS y Android
- Arquitectura del renderizador
- Anotaciones

TsgcPDFCommentsViewFMX

TsgcPDFCommentsViewFMX (unidad sgcPDF_ViewerFMX) es la versión FireMonkey de TsgcPDFCommentsView. Es un TTreeView que enumera los comentarios del documento de un TsgcPDFViewerFMX. Se enumeran todas las anotaciones excepto los campos de formulario, los enlaces y los popups. Requiere Delphi XE5 o posterior y lo instalan los paquetes FMX, en la página de paleta **sgcPDF**.

Miembro	Descripción
Viewer: TsgcPDFViewerFMX	El visor que se sigue. La lista se reconstruye cuando el visor carga o cierra un documento y cuando cambian sus anotaciones.
GroupByPage: Boolean	Enumera los comentarios bajo un elemento por página, con el texto Page N (count). Cuando es False, los comentarios forman una lista plana en el orden de las páginas. Valor predeterminado True.
ShowReplies: Boolean	Muestra las respuestas de un comentario como sus elementos hijos. Cuando es False, solo se enumeran los comentarios de nivel superior. Valor predeterminado True.
RebuildComments	Vuelve a cargar la lista a partir del documento del visor. La lista ya sigue al visor, y después de modificar Document.Annotations en código el método AnnotationsChanged del visor también la reconstruye.
SelectedAnnotation: TsgcPDFAnnotation	Solo lectura. La anotación del elemento seleccionado, nil cuando no hay ningún elemento seleccionado o es un elemento de página.

Cada elemento muestra `Author, date: contents`. El contenido se muestra en una línea, y los textos largos se acortan. Cuando el contenido está vacío, se muestra el tipo de la anotación, como Highlight. Una anotación sin autor muestra (no author), y la fecha se omite cuando la anotación no tiene ninguna.

- Al seleccionar un elemento, el visor se desplaza hasta la anotación con GotoAnnotation. Con AnnotationEditing del visor en True, la anotación también se selecciona en el visor.
- Un doble clic abre el popup de la anotación.
- La tecla Supr elimina la anotación seleccionada y sus respuestas mientras AnnotationEditing del visor sea True. OnAnnotationDelete del visor puede rechazarlo.
- Cuando el usuario selecciona una anotación en el visor, su elemento se selecciona en la lista.

Las demás propiedades y eventos, como Align, Width y OnChange, son los del TTreeView de FMX. Los elementos se crean a partir del documento y no se guardan en el formulario.

Ejemplo: panel de comentarios

```
procedure TMainForm.CreateCommentsPanel;
begin
  FComments := TsgcPDFCommentsViewFMX.Create(Self);
  FComments.Parent := Self;
  FComments.Align := TAlignLayout.Right;
  FComments.Width := 280;
  FComments.GroupByPage := True;
  FComments.ShowReplies := True;
  FComments.OnChange := OnCommentChangeEvent;
  // from now on the list follows the document, the annotations and the
  // selection of the viewer
  FComments.Viewer := FViewer;
  // a selected comment is also selected in the viewer, Delete removes it
  FViewer.AnnotationEditing := True;
end;

procedure TMainForm.OnCommentChangeEvent(Sender: TObject);
var
```

```
  oAnnot: TsgcPDFAnnotation;
begin
  // nil for a page group item
  oAnnot := FComments.SelectedAnnotation;
  if oAnnot <> nil then
    Caption := Format('%s, page %d', [oAnnot.Author, oAnnot.PageIndex + 1]);
end;
```

Vea también

- TsgcPDFViewerFMX
- TsgcPDFCommentsView
- Anotaciones

Fuentes OpenType

`TsgcPDFDocument.AddTrueTypeFont` (una fuente instalada) y `AddTrueTypeFontFromFile` (un archivo .ttf, .otf o la primera fuente de un archivo .ttc) aceptan ambas variantes de fuentes OpenType:

Contornos de fuente	Incrustado como
TrueType (tabla glyf, archivos .ttf y la mayoría de archivos .otf)	Fuente Type0 con un descendiente CIDFontType2 y un programa FontFile2 de subconjunto.
PostScript / CFF (tabla CFF, archivos .otf que comienzan con OTTO)	Fuente Type0 con un descendiente CIDFontType0 y un programa FontFile3 de subconjunto del subtipo CIDFontType0C.

En ambos casos solo se incrustan los glifos utilizados en el documento, se escribe un CMap ToUnicode para que el texto se pueda copiar y buscar, y los anchos de glifo se escriben en /W. `TsgcTrueTypeFont.IsCFF` indica qué variante tiene una fuente cargada. El texto se dibuja con `TsgcPDFPage.DrawTextTT` o `DrawTextTTColor`, que también aplican las características de composición OpenType descritas en Escrituras complejas y shaping.

Ejemplo

```
procedure CreateOpenTypeCFFPDF(const AOTFFile, AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vFont: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    // an .otf font with PostScript (CFF) outlines is embedded as a
    // CIDFontType0 font with a subset FontFile3 /CIDFontType0C program
    vFont := oPDF.AddTrueTypeFontFromFile(AOTFFile);
    oPage.DrawTextTT('OpenType CFF text, subset and embedded', 50, 780,
      vFont, 16);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

AddTrueTypeFontFromFile es la forma de utilizar fuentes en servidores, en Linux y en plataformas móviles, donde es posible que la fuente no esté instalada. `FindTrueTypeFont(AFamilyName, ABold, AItalic)` devuelve el índice de una fuente ya agregada, o -1.

Vea también

- Escrituras complejas y shaping
- Texto en chino, japonés y coreano
- Texto y fuentes

Escrituras complejas y shaping

El texto dibujado con una fuente TrueType u OpenType incrustada (`DrawTextTT`, `DrawTextTTColor`, TsgcPDFCanvas y TsgcPDFHTMLRenderer con fuentes TrueType, y las apariencias de campos de formulario) pasa por el shaper OpenType de sgcPDF (unidad sgcPDF_Shaping, clase TsgcPDFShaper). Está escrito en Pascal puro y funciona en todas las plataformas. Usted pasa el texto en orden lógico, tal como se escribe, y el shaper produce los glifos en orden visual:

- **Árabe**: formas de unión (isol, init, medi, fina), ligaduras obligatorias (rlig) como lam alef, y posicionamiento de marcas.
- **Hebreo** y otro texto de derecha a izquierda, con caracteres reflejados como los corchetes.
- **Texto bidireccional**: el algoritmo bidireccional de Unicode (UAX #9) ordena los fragmentos mixtos de izquierda a derecha y de derecha a izquierda, números incluidos. La dirección del párrafo procede del primer carácter fuerte.
- **Ligaduras y formas contextuales**: ccmp, locl, liga, clig y calt de la tabla GSUB.
- **Kerning y colocación de marcas**: kern, mark, mkmk y curs de la tabla GPOS, o la tabla kern heredada cuando la fuente no tiene kerning GPOS.

Cuando el texto contiene fragmentos de derecha a izquierda, el texto lógico también se escribe como /ActualText, de modo que copiar y pegar y la extracción de texto devuelven el texto en el orden en que se escribió.

Ejemplo: árabe, hebreo y kerning

```
procedure CreateMultilingualPDF(const AFontFile, AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vFont: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    // a TrueType or OpenType font file; the used glyphs are embedded
    vFont := oPDF.AddTrueTypeFontFromFile(AFontFile);
    // Latin with kerning and ligatures from the GPOS / GSUB tables
    oPage.DrawTextTT('Office affinity, AVATAR', 50, 780, vFont, 18);
    // Arabic: joining forms and right to left order are applied
    oPage.DrawTextTT(WideString(#$0645#$0631#$062D#$0628#$0627), 50, 740,
      vFont, 18);
    // Hebrew mixed with numbers (Unicode bidirectional algorithm)
    oPage.DrawTextTT(WideString(#$05E9#$05DC#$05D5#$05DD) + ' 2026', 50, 700,
      vFont, 18);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Usar el shaper directamente

`TsgcTrueTypeFont.ShapeText(const AText: WideString; ABaseLevel: Integer = -1): TsgcPDFShapedGlyphs` devuelve los glifos posicionados (GlyphID, Cluster, Text, XAdvance, YAdvance, XOffset, YOffset y el Level bidi) en orden visual, con los avances en unidades de fuente. ABaseLevel es 0 para izquierda a derecha, 1 para derecha a izquierda y -1 para detectarlo. `EncodeShapedText(AText, AFontSize)` devuelve los operadores del stream de contenido para un texto con shaping. La unidad sgcPDF_Shaping también expone las funciones auxiliares bidi `sgcPDFBidiVisual`, `sgcPDFBidiLevels` y `sgcPDFBidiReorder`.

```
procedure ShowShapedGlyphs(const AText: WideString);
var
  oFont: TsgcTrueTypeFont;
  vGlyphs: TsgcPDFShapedGlyphs;
  i: Integer;
```

```
begin
  oFont := TsgcTrueTypeFont.Create;
  try
    oFont.LoadFromFile('C:\Windows\Fonts\arial.ttf');
    // -1: the paragraph direction comes from the first strong character
    vGlyphs := oFont.ShapeText(AText, -1);
    for i := 0 to Length(vGlyphs) - 1 do
      WriteLn(Format('glyph %d  cluster %d  advance %d  level %d',
        [vGlyphs[i].GlyphID, vGlyphs[i].Cluster, vGlyphs[i].XAdvance,
        vGlyphs[i].Level]));
  finally
    oFont.Free;
  end;
end;
```

Campos de formulario

Los valores de campo de formulario con caracteres fuera de la codificación de fuente estándar (cirílico, griego, CJK, árabe, hebreo) obtienen apariencias dibujadas con una fuente TrueType incrustada y con el mismo shaping. Establezca `TsgcPDFAcroForm.AppearanceFontFile` o `AppearanceFontName` para elegir la fuente. Cuando ambos están vacíos, se utiliza la primera fuente instalada que cubra los caracteres (Arial Unicode MS, Segoe UI, Arial, Yu Gothic, MS Gothic, Microsoft YaHei, Malgun Gothic, Noto Sans).

Limitaciones

- Las escrituras que necesitan un motor de shaping dedicado, como las escrituras índicas (devanagari, bengalí, tamil) y del sudeste asiático (tailandés, jemer), se dibujan solo con las características genéricas, sin reordenación de sílabas.
- Los caracteres explícitos de incrustación y anulación bidi (LRE, RLE, LRO, RLO, isolates) se ignoran.

Vea también

- Fuentes OpenType
- Texto en chino, japonés y coreano
- Formularios

Texto en chino, japonés y coreano

Para escribir texto en chino, japonés o coreano, agregue una fuente que contenga los caracteres con `AddTrueTypeFont` o `AddTrueTypeFontFromFile` y dibuje con `DrawTextTT`. La fuente se incrusta como un subconjunto (solo los glifos utilizados), de modo que la salida se mantiene pequeña incluso con fuentes CJK grandes, y se muestra igual en todos los equipos. Se aceptan colecciones TrueType (.ttc): se utiliza la primera fuente de la colección.

```
procedure CreateJapanesePDF(const AFontFile, AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vFont: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    // for example C:\Windows\Fonts\msgothic.ttc (the first face is used)
    vFont := oPDF.AddTrueTypeFontFromFile(AFontFile);
    oPage.DrawTextTT(WideString(#$65E5#$672C#$8A9E#$306E#$30C6#$30AD#$30B9#$30C8),
      50, 780, vFont, 20);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Fuentes típicas de Windows: MS Gothic, Yu Gothic y Meiryo (japonés), SimSun y Microsoft YaHei (chino simplificado), MingLiU y Microsoft JhengHei (chino tradicional), Batang, Gulim y Malgun Gothic (coreano). En Linux utilice, por ejemplo, las fuentes Noto Sans CJK o WenQuanYi con AddTrueTypeFontFromFile.

Lectura y renderizado de documentos CJK

La extracción de texto y los renderizadores admiten documentos CJK creados por otras herramientas: fuentes compuestas con codificaciones Identity-H e Identity-V y los CMaps CJK predefinidos, fuentes CID incrustadas (TrueType y CFF), y escritura vertical. Los documentos que utilizan fuentes CJK sin incrustarlas se renderizan con una fuente instalada del sistema.

TsgcPDFCJKFontManager

La unidad sgcPDF_Fonts_CJK (solo Windows) conserva las clases de bajo nivel más antiguas TsgcPDFCIDFont y TsgcPDFCJKFontManager y las funciones auxiliares `sgcPDFIsCJKCodePoint` y `TsgcPDFCJKFontManager.ContainsCJK`. Estas clases no están conectadas a TsgcPDFDocument. Utilice AddTrueTypeFont y DrawTextTT para código nuevo.

Vea también

- Fuentes OpenType
- Escrituras complejas y shaping

Extracción de texto: posiciones de caracteres

Además del texto plano y los fragmentos, **TsgcPDFTextExtractor** (unidad sgcPDF_TextExtract) devuelve cada carácter con su posición, agrupados en palabras y líneas en orden de lectura. Esto es lo que utiliza el visor VCL para la selección de texto y el resaltado de búsqueda, y puede utilizarlo para la búsqueda de texto con coordenadas, la extracción de datos de diseños fijos o la redacción del texto encontrado.

API de caracteres, palabras y líneas

Método	Descripción
GetCharCount, GetChar(AIndex): TsgcPDFTextChar	Caracteres en el orden de lectura de GetText. TsgcPDFTextChar tiene el cuadrilátero del glifo (X1, Y1 a X4, Y4, siguiendo la dirección del texto), el cuadro delimitador (Left, Bottom, Right, Top), el origen (OriginX, OriginY), FontSize, LineIndex, WordIndex (-1 para espacio en blanco) y Synthetic (un espacio insertado por el extractor entre dos fragmentos de texto).
GetWordCount, GetWord(AIndex): TsgcPDFTextWord	Palabras como un rango de caracteres (FirstChar, CharCount), con LineIndex y cuadro delimitador.
GetLineCount, GetLine(AIndex): TsgcPDFTextLine	Líneas como un rango de caracteres con su cuadro delimitador.
GetCharsText: WideString	Los caracteres unidos con CRLF entre líneas. Mismo texto que GetText.
CharAt(AX, AY: Double; ATolerance: Double = 0): Integer	Índice del carácter en un punto del espacio de usuario, o -1. ATolerance amplía los cuadros de los caracteres.
IncludeAnnotations: Boolean	También extrae el texto mostrado por los streams de apariencia de anotaciones visibles (valores de campos de formulario, texto libre y sellos). Predeterminado False.

`sgcPDFCreatePageTextExtractor(AParser, APageIndex)` crea un extractor para una página y ejecuta Extract (usted lo libera, nil cuando la página no existe). Todas las posiciones están en el espacio de usuario de PDF: puntos, origen en la esquina inferior izquierda.

Ejemplo

```
procedure ListCharacterPositions(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oText: TsgcPDFTextExtractor;
  vChar: TsgcPDFTextChar;
  vWord: TsgcPDFTextWord;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oText := TsgcPDFTextExtractor.Create(oPDF.Parser, oPDF.Pages[0].PageDict);
    try
```

```
      // also read the text shown by annotation appearance streams
      oText.IncludeAnnotations := True;
      oText.Extract;
      for i := 0 to oText.GetWordCount - 1 do
      begin
        vWord := oText.GetWord(i);
        WriteLn(Format('word %d: line %d, box %.1f %.1f %.1f %.1f',
          [i, vWord.LineIndex, vWord.Left, vWord.Bottom, vWord.Right,
          vWord.Top]));
      end;
      if oText.GetCharCount > 0 then
      begin
        vChar := oText.GetChar(0);
        WriteLn(Format('first char at %.1f, %.1f size %.1f',
          [vChar.OriginX, vChar.OriginY, vChar.FontSize]));
        // hit test: the character under a point of the page (user space)
        WriteLn('char at point: ', oText.CharAt(vChar.OriginX + 1,
          vChar.OriginY + 2, 1));
      end;
    finally
      oText.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Extracción mejorada

- El texto dentro de los XObjects de formulario se extrae, con la matriz del formulario aplicada a las posiciones.
- El /ActualText del contenido marcado reemplaza a los glifos que cubre, de modo que las ligaduras, el árabe con shaping y las marcas de guionado se extraen como el texto original.
- Las fuentes sin un CMap ToUnicode se mapean mediante su codificación, /Differences y nombres de glifo.

Vea también

- TsgcPDFTextExtractor
- Extracción de texto
- TsgcPDFViewerControl

Límites para archivos no confiables

sgcPDF se utiliza a menudo para abrir archivos procedentes del exterior: cargas, adjuntos de correo electrónico, documentos escaneados. Cada analizador y decodificador comprueba su entrada y genera una excepción ante datos dañados en lugar de leer fuera de los límites. Las asignaciones están limitadas por estas variables globales, que puede reducir (o aumentar) antes de cargar documentos:

Variable	Unidad	Predeterminado	Límite
sgcPDFMaxDecodedSize: Int64	sgcPDF_Filters	268435456 (256 MB)	Mayor tamaño decodificado de un stream (Flate, LZW, RunLength, ASCII85, ASCIIHex, CCITT, JBIG2). Protege contra bombas de descompresión.
sgcPDFJPEGMaxPixels: Int64	sgcPDF_JPEG	67108864	Mayor fotograma JPEG (ancho x alto).
sgcPDFJPEGMaxMemory: Int64	sgcPDF_JPEG	1073741824 (1 GB)	Memoria de trabajo del decodificador JPEG para una imagen.
sgcPDFJPXMaxPixels: Int64	sgcPDF_JPX	67108864	Mayor imagen JPEG 2000 (ancho x alto).
CS_CANVAS_MAX_RASTER (constante)	sgcPDF_Renderer_Canvas	67108864	Mayor ráster construido por los renderizadores, en píxeles.

```
procedure ConfigureLimits;
begin
  // largest decoded stream (default 256 MB); larger streams raise
  sgcPDFMaxDecodedSize := 64 * 1024 * 1024;
end;
```

Hay otras protecciones incorporadas: límites de recursión para XObjects de formulario anidados, patrones, glifos Type 3, árboles de páginas y esquemas, detección de ciclos en las referencias de objetos, y cachés acotadas en los visores. Una imagen dañada se omite o se dibuja parcialmente, y el resto de la página se sigue renderizando. Estas variables son globales al proceso. Establézcalas una sola vez al iniciar, no mientras otros hilos están cargando documentos.

Vea también

- Gestión de errores
- Decodificadores de imagen

Decodificadores de imagen

sgcPDF incluye decodificadores en Pascal puro para todas las compresiones de imagen utilizadas en archivos PDF. Los renderizadores, los visores, TsgcPDFImageExtractor y el optimizador los utilizan, y usted puede llamarlos directamente. Se ejecutan en todas las plataformas y no necesitan ninguna DLL.

Formato	Unidad y función	Admitido
JPEG (DCTDecode)	sgcPDF_JPEG: sgcPDFDecodeJPEG, TsgcPDFJPEGDecoder	Huffman baseline y progresivo, 8 bits, 1, 3 o 4 componentes, cualquier factor de muestreo, marcadores de reinicio, transformaciones JFIF y Adobe (YCbCr, YCCK, CMYK). La codificación aritmética, JPEG sin pérdida y de 12 bits generan EsgcPDFJPEGError.
JPEG 2000 (JPXDecode)	sgcPDF_JPX: sgcPDFDecodeJPX	Archivos JP2 y codestreams sin procesar, mosaicos, todos los órdenes de progresión, wavelets reversibles 5/3 e irreversibles 9/7, submuestreo de componentes, de 1 a 16 bits, paletas, canales alfa, sYCC.
CCITT Group 3 and Group 4 (CCITTFaxDecode)	sgcPDF_CCITT: sgcPDFCCITTFaxDecode, sgcPDFCCITTFaxDecodeParms	Group 3 unidimensional (MH), Group 3 bidimensional (MR) y Group 4 (MMR), EndOfLine, EncodedByteAlign, EndOfBlock, BlackIs1.
JBIG2 (JBIG2Decode)	sgcPDF_JBIG2: sgcPDFJBIG2Decode	Regiones genéricas (aritméticas y MMR), diccionarios de símbolos y regiones de texto con codificación aritmética, /JBIG2Globals. Los segmentos de medio tono, patrón, refinamiento y símbolo y texto codificados con Huffman no se admiten y generan una excepción.

Firmas

- `sgcPDFDecodeJPEG(const AData: TBytes; out AWidth, AHeight, AComponents: Integer): TBytes` y una sobrecarga con `AColorTransform: Integer` (el valor /ColorTransform del PDF, -1 cuando está ausente). Devuelve muestras de 8 bits entrelazadas.
- `sgcPDFDecodeJPX(const AData: TBytes; out AInfo: TsgcPDFJPXInfo; AKeep16: Boolean = False): TBytes`. AInfo proporciona Width, Height, Components, ColorComponents, HasAlpha, BitsPerComponent, ColorSpace (sgcJPXColorGray, sgcJPXColorRGB, sgcJPXColorCMYK, sgcJPXColorICC) y el perfil ICC.
- `sgcPDFCCITTFaxDecode(const AData: TBytes; AK, AColumns, ARows: Integer; AEndOfLine, AEncodedByteAlign, AEndOfBlock, ABlackIs1: Boolean; ADamagedRowsBeforeError: Integer = 0): TBytes`. AK < 0 es Group 4, 0 es Group 3 MH, > 0 es Group 3 MR. Devuelve filas de 1 bit rellenadas hasta un byte.
- `sgcPDFJBIG2Decode(const AData, AGlobals: TBytes; out AWidth, AHeight: Integer): TBytes`. Devuelve filas de 1 bit con la convención de PDF (0 es negro).

Ejemplo

```
procedure DecodeJPEGFile(const AFileName: string);
var
  vPixels: TBytes;
  vWidth, vHeight, vComponents: Integer;
begin
  // 1 (gray), 3 (RGB) or 4 (CMYK) bytes per pixel
  vPixels := sgcPDFDecodeJPEG(ReadFileBytes(AFileName), vWidth, vHeight,
    vComponents);
  WriteLn(Format('%d x %d, %d components', [vWidth, vHeight, vComponents]));
end;
```

Todos los decodificadores rechazan los datos dañados con una excepción y limitan sus asignaciones, vea Límites para archivos no confiables.

Vea también

- TsgcPDFImageExtractor
- Importación de TIFF
- Características de renderizado admitidas

Importación de TIFF

TsgcPDFImage y TsgcPDFDocument.AddImage cargan archivos TIFF además de JPEG, PNG y BMP. Esto facilita convertir documentos escaneados y archivos de fax en PDF.

Archivos TIFF admitidos

- Little endian y big endian (II y MM), tiras y mosaicos.
- Compresión: ninguna, PackBits, LZW, Deflate (con predictor horizontal), CCITT Group 3 y Group 4, y JPEG.
- Imágenes bilevel, en escala de grises, RGB, con paleta y CMYK, de 1 a 16 bits por muestra (16 bits se reducen a 8).
- Archivos multipágina: se puede agregar cada página (IFD).

Las páginas CCITT y JPEG de una sola tira se incrustan sin recodificar, como streams CCITTFaxDecode y DCTDecode, de modo que una página de fax escaneada conserva su tamaño y calidad originales. Las demás páginas se comprimen con Flate.

API

Miembro	Descripción
TsgcPDFDocument.AddImage(const AFileName: string; APageIndex: Integer): Integer	Agrega la página APageIndex (con base cero) de un archivo TIFF multipágina. Los demás formatos solo aceptan la página 0.
TsgcPDFImage.TIFFPageCount(const AData: TBytes): Integer	Función de clase: número de páginas de un archivo TIFF.
TsgcPDFImage.LoadTIFF(const AData: TBytes; APageIndex: Integer = 0)	Carga una página en un TsgcPDFImage (luego agréguela con TsgcPDFDocument.AddImageObject).
TsgcPDFImage.IsCCITT: Boolean	True cuando la página se incrusta como datos CCITT sin recodificar.

Ejemplo: TIFF multipágina a PDF

```
procedure TIFFToPDF(const ATIFFFile, APDFFile: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vPages, vImage, i: Integer;
begin
  // number of pages (IFDs) of the TIFF file
  vPages := TsgcPDFImage.TIFFPageCount(ReadFileBytes(ATIFFFile));
  oPDF := TsgcPDFDocument.Create;
  try
    for i := 0 to vPages - 1 do
    begin
      oPage := oPDF.AddPage(595, 842);
      // APageIndex selects the TIFF page; CCITT G3/G4 and JPEG strips are
      // embedded without re-encoding
      vImage := oPDF.AddImage(ATIFFFile, i);
      oPage.DrawImageFromIndex(vImage, 50, 300, 495, 371);
    end;
    oPDF.SaveToFile(APDFFile);
  finally
    oPDF.Free;
```

```
  end;
end;
```

Vea también

- Imágenes
- Decodificadores de imagen

TsgcPDFImageExtractor

TsgcPDFImageExtractor (unidad sgcPDF_ImageExtract) enumera las imágenes de un PDF y las guarda como archivos PNG, JPEG o BMP, o devuelve sus píxeles. Encuentra XObjects de imagen, imágenes en línea e imágenes dentro de XObjects de formulario (y opcionalmente dentro de anotaciones), con la posición en la que se pinta cada imagen. Está en la página de paleta sgcPDF.

Propiedades

Propiedad	Descripción
Document: TsgcPDFDocument	Un documento cargado.
Parser: TsgcPDFParser	Un analizador explícito. Tiene prioridad sobre Document.
IncludeInline: Boolean	Enumera las imágenes en línea (BI, ID, EI). Predeterminado True.
IncludeAnnotations: Boolean	Enumera las imágenes pintadas por streams de apariencia de anotaciones. Predeterminado False.
IncludeUnused: Boolean	Enumera las imágenes de los recursos de página que el contenido nunca pinta. Predeterminado True.
UniqueOnly: Boolean	Enumera cada objeto de imagen una sola vez. UseCount indica cuántas veces se pinta. Predeterminado False.
IncludeAlpha: Boolean	Agrega la transparencia /SMask o /Mask como un canal alfa. Predeterminado True.
MaxFormDepth: Integer	Anidamiento máximo de XObjects de formulario. Predeterminado 16.
FileNamePattern: string	Nombre de archivo utilizado por ExtractAll, sin extensión: %p número de página, %i número de imagen, %n número de imagen dentro de la página, %o número de objeto, %r nombre de recurso.
Count, Items[AIndex]: TsgcPDFImageItem	Las imágenes encontradas. El escaneo se ejecuta bajo demanda.
Errors: TStringList	Imágenes que ExtractAll no pudo guardar.

Métodos

Método	Descripción
Scan, ScanPage(APageIndex), Clear	Enumera las imágenes de todas las páginas o de una página.
SaveToFile(AIndex; const AFileName: string)	Elige el formato a partir de la extensión: .png, .jpg, .jpeg, .bmp. Las extensiones .jp2, .jpx, .j2k, .jb2, .bin y .raw guardan los datos del stream tal como se almacenan.
SaveAsPNG, SaveAsJPEG, SaveAsBMP, SaveToStream(AIndex; AStream; AFormat)	Guarda una imagen. AFormat: pifAuto (las imágenes JPEG como JPEG, JPEG 2000 tal como se almacena, el resto como PNG), pifPNG, pifJPEG, pifBMP o pifRaw.

Método	Descripción
ExtractAll(const AFolder: string; AFormat = pifAuto): Integer	Guarda todas las imágenes enumeradas y devuelve el número de archivos escritos.
GetPixels(AIndex; out AWidth, AHeight, AChannels: Integer): TBytes	Píxeles decodificados de 8 bits, de arriba hacia abajo: 1 Gray, 2 Gray y alfa, 3 RGB, 4 RGBA.
GetRawBytes, GetJPEGBytes, GetPNGBytes, GetBMPBytes, CanSaveAsJPEG	Datos de imagen en memoria. La salida JPEG está disponible para imágenes DCTDecode, que se guardan sin recompresión.

TsgcPDFImageItem describe una imagen: PageIndex, ObjectNumber, Name, Source (pisXObject o pisInline), Width, Height, BitsPerComponent, ColorSpace, Filters, Matrix, Rect (el área pintada en puntos), Placed, InAnnotation, HasSMask, HasMask, IsImageMask, UseCount y Stream.

Ejemplo: enumerar y guardar imágenes

```
procedure ExtractImages(const APDFFile, AFolder: string);
var
  oPDF: TsgcPDFDocument;
  oExtractor: TsgcPDFImageExtractor;
  oItem: TsgcPDFImageItem;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    oExtractor := TsgcPDFImageExtractor.Create(nil);
    try
      oExtractor.Document := oPDF;
      oExtractor.UniqueOnly := True;
      for i := 0 to oExtractor.Count - 1 do
      begin
        oItem := oExtractor.Items[i];
        WriteLn(Format('page %d: %dx%d %s [%s] at %.0f,%.0f', [oItem.PageIndex +
          1, oItem.Width, oItem.Height, oItem.ColorSpace, oItem.Filters,
          oItem.Rect.Left, oItem.Rect.Bottom]));
      end;
      // one file per image: page number and image number in the name
      oExtractor.FileNamePattern := 'page%p_img%n';
      WriteLn('files written: ', oExtractor.ExtractAll(AFolder, pifPNG));
      // or a single image in the format given by the extension
      if oExtractor.Count > 0 then
        oExtractor.SaveToFile(0, IncludeTrailingPathDelimiter(AFolder) +
          'first.bmp');
    finally
      oExtractor.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Ejemplo: píxeles

```
procedure ReadImagePixels(const APDFFile: string);
var
  oPDF: TsgcPDFDocument;
  oExtractor: TsgcPDFImageExtractor;
  vPixels: TBytes;
  vWidth, vHeight, vChannels: Integer;
```

```
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    oExtractor := TsgcPDFImageExtractor.Create(nil);
    try
      oExtractor.Document := oPDF;
      oExtractor.ScanPage(0);
      if oExtractor.Count > 0 then
      begin
        // decoded samples, 8 bits per channel, top-down rows
        vPixels := oExtractor.GetPixels(0, vWidth, vHeight, vChannels);
        WriteLn(Format('%d x %d, %d channels, %d bytes', [vWidth, vHeight,
          vChannels, Length(vPixels)]));
      end;
    finally
      oExtractor.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Los codificadores también están disponibles como funciones: `sgcPDFEncodePNG` y `sgcPDFEncodeBMP(const APixels: TBytes; AWidth, AHeight, AChannels: Integer): TBytes`.

Vea también

- Decodificadores de imagen
- Imágenes

TsgcPDFCanvas

TsgcPDFCanvas (unidad sgcPDF_Canvas) dibuja en una página PDF con una API modelada según el TCanvas de la VCL: objetos Pen, Brush y Font, MoveTo y LineTo, Rectangle, Ellipse, Arc, Pie, TextOut, TextRect, etc. Todo se escribe como gráficos vectoriales y texto real en el contenido de la página. Es la forma más sencilla de portar a PDF código de impresión con TCanvas existente, y es multiplataforma (sin VCL, sin GDI).

Coordenadas

El origen del canvas es la esquina **superior izquierda** de la página y el eje Y crece hacia abajo, como en un TCanvas. Elija la unidad en el constructor: `Create(APage: TsgcPDFPage; AUnit: TsgcPDFCanvasUnit = pcuPoints; ADPI: Double = 96)`, con pcuPoints, pcuPixels (a ADPI), pcuMillimeters o pcuInches. Translate, Rotate, Scale, MultiplyTransform, SetTransform y ResetTransform cambian la transformación.

Miembros principales

Miembro	Descripción
Pen	Color, Width (0 dibuja la línea visible más fina), Style (ppsSolid, ppsDash, ppsDot, ppsDashDot, ppsDashDotDot, ppsClear), Cap, Join, MiterLimit, GapFill y GapColor.
Brush	Color y Style (pbsSolid, pbsClear).
Font	Name, Size (puntos), Style (pfsBold, pfsItalic, pfsUnderline, pfsStrikeOut) y Color.
Opacity: Double	Alfa de relleno y trazo, de 0 a 1.
TrueTypeFonts: Boolean	Las fuentes distintas de Arial/Helvetica, Times y Courier se incrustan como TrueType cuando están instaladas. False asigna todas las fuentes a las 14 fuentes estándar.
MoveTo, LineTo, Rectangle, FillRect, FrameRect, RoundRect, Ellipse, Arc, ArcTo, Pie, Chord, Polygon, Polyline, PolylineTo, PolyBezier, PolyBezierTo	Líneas y formas, con las convenciones de parámetros de TCanvas.
BeginPath, EndPath, CloseFigure, FillPath, StrokePath, StrokeAndFillPath, ClipPath	Trayectos.
IntersectClipRect, ExcludeClipRect, ResetClip, SaveState, RestoreState	Recorte y estado.
TextOut, TextOutEx, TextRect, TextWidth, TextHeight, TextAscent, TextDescent	Texto, con alineación, ajuste de línea, avances de carácter y rotación (TextOutEx).
DrawImage(X, Y, W, H; AImageIndex | AData: TBytes | AStream | AImage: TsgcPDFImage)	Imágenes.
EndDraw	Cierra el grupo de recorte abierto. Llámelo al terminar de dibujar una página.

Los colores son valores TsgcPDFCanvasColor con el formato de TColor ($00BBGGRR). Constrúyalos con `sgcPDFCanvasRGB(R, G, B)`.

Ejemplo

```
procedure DrawWithCanvas(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oCanvas: TsgcPDFCanvas;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // millimeters, origin at the top left corner, Y grows downwards
    oCanvas := TsgcPDFCanvas.Create(oPDF.AddPage(595, 842), pcuMillimeters);
    try
      oCanvas.Pen.Color := sgcPDFCanvasRGB(0, 64, 128);
      oCanvas.Pen.Width := 0.5;
      oCanvas.Brush.Color := sgcPDFCanvasRGB(220, 235, 250);
      oCanvas.RoundRect(20, 20, 190, 60, 6, 6);
      oCanvas.Font.Name := 'Arial';
      oCanvas.Font.Size := 18;
      oCanvas.Font.Style := [pfsBold];
      oCanvas.TextRect(20, 20, 190, 60, 'Drawn with TsgcPDFCanvas', ptaCenter,
        ptvCenter);
      oCanvas.Brush.Color := sgcPDFCanvasRGB(255, 200, 0);
      oCanvas.Pie(20, 80, 80, 140, 80, 110, 50, 80);
      oCanvas.Pen.Style := ppsDash;
      oCanvas.MoveTo(100, 80);
      oCanvas.LineTo(190, 140);
      // semi transparent overlay
      oCanvas.Opacity := 0.4;
      oCanvas.Brush.Color := sgcPDFCanvasRGB(255, 0, 0);
      oCanvas.Ellipse(120, 90, 170, 130);
      oCanvas.EndDraw;
    finally
      oCanvas.Free;
    end;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- EMF a PDF
- TsgcPDFVCLCanvasHelper
- HTML a PDF
- Gráficos

EMF a PDF

TsgcPDFEMFPlayer (unidad sgcPDF_EMF, solo Windows) reproduce metarchivos mejorados de Windows (EMF) en un TsgcPDFCanvas. Las líneas, formas, trayectos, texto y mapas de bits del metarchivo se convierten en gráficos vectoriales, texto real e imágenes en el PDF, de modo que los gráficos e informes dibujados por otros componentes se mantienen nítidos a cualquier zoom y el texto se puede buscar.

API

Miembro	Descripción
Create(ACanvas: TsgcPDFCanvas)	El canvas que recibe el dibujo.
Play(AHandle: HENHMETAFILE; X, Y, W, H: Double)	Reproduce un metarchivo de modo que su marco llene el rectángulo (unidades del canvas).
PlayFromFile, PlayFromStream, PlayFromBytes	Igual, desde un archivo .emf, un stream o bytes.
GetFrameSize(AHandle; out AWidth, AHeight: Double)	Procedimiento de clase: tamaño del marco del metarchivo en puntos.
RecordCount, SkippedRecords	Número de registros reproducidos y de registros no admitidos.
sgcPDFEMFAddPage(ADocument; AHandle | AFileName): TsgcPDFPage	Agrega una página con el tamaño del marco del metarchivo y dibuja el metarchivo en ella.

Ejemplo

```
procedure EMFToPDF(const AEMFFile, APDFFile: string);
var
  oPDF: TsgcPDFDocument;
  oCanvas: TsgcPDFCanvas;
  oPlayer: TsgcPDFEMFPlayer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // a page with the size of the metafile frame
    sgcPDFEMFAddPage(oPDF, AEMFFile);
    // or play a metafile into a rectangle of any page
    oCanvas := TsgcPDFCanvas.Create(oPDF.AddPage(595, 842));
    try
      oPlayer := TsgcPDFEMFPlayer.Create(oCanvas);
      try
        oPlayer.PlayFromFile(AEMFFile, 50, 50, 300, 225);
        WriteLn(Format('records %d, skipped %d', [oPlayer.RecordCount,
          oPlayer.SkippedRecords]));
      finally
        oPlayer.Free;
      end;
      oCanvas.EndDraw;
    finally
      oCanvas.Free;
    end;
    oPDF.SaveToFile(APDFFile);
  finally
    oPDF.Free;
```

```
  end;
end;
```

Vea también

- TsgcPDFCanvas
- TsgcPDFVCLCanvasHelper

TsgcPDFVCLCanvasHelper

TsgcPDFVCLCanvasHelper (unidad sgcPDF_CanvasVCL) conecta los gráficos VCL con un TsgcPDFCanvas. Solo Windows y VCL.

Miembro	Descripción
Create(ACanvas: TsgcPDFCanvas)	El canvas de destino.
DrawMetafile(AMetafile: TMetafile; X, Y, W, H: Double)	Reproduce un TMetafile como gráficos vectoriales.
DrawGraphic(AGraphic: TGraphic; X, Y, W, H: Double)	Dibuja cualquier TGraphic: metarchivos como gráficos vectoriales, mapas de bits e imágenes PNG y JPEG como imágenes.
DrawControl(AControl: TWinControl; X, Y, W, H: Double)	Dibuja un control y sus hijos (por ejemplo un gráfico o una cuadrícula) como gráficos vectoriales, mediante un metarchivo mejorado.
ControlToMetafile(AControl: TWinControl): TMetafile	Función de clase: graba un control en un nuevo metarchivo (usted lo libera).

X, Y, W y H están en unidades del canvas.

Ejemplo

```
procedure GraphicToPDF(AGraphic: TGraphic; const APDFFile: string);
var
  oPDF: TsgcPDFDocument;
  oCanvas: TsgcPDFCanvas;
  oHelper: TsgcPDFVCLCanvasHelper;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oCanvas := TsgcPDFCanvas.Create(oPDF.AddPage(595, 842));
    try
      oHelper := TsgcPDFVCLCanvasHelper.Create(oCanvas);
      try
        // a TMetafile is played as vector graphics, other graphics as images
        oHelper.DrawGraphic(AGraphic, 50, 50, 400, 300);
      finally
        oHelper.Free;
      end;
      oCanvas.EndDraw;
    finally
      oCanvas.Free;
    end;
    oPDF.SaveToFile(APDFFile);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFCanvas
- EMF a PDF

HTML a PDF

TsgcPDFHTMLRenderer (unidad sgcPDF_HTML) convierte HTML con CSS en páginas PDF paginadas. Tiene su propio motor de maquetación escrito en Pascal puro, por lo que no necesita ningún navegador y se ejecuta en todas las plataformas. Utilícelo para facturas, informes, cartas y archivos de correo electrónico generados a partir de plantillas HTML. Está en la página de paleta sgcPDF.

Métodos

Método	Descripción
HTMLToPDFFile(const AHTML, APDFFileName: string; const ABasePath: string = '')	Crea un archivo PDF a partir de una cadena HTML.
HTMLFileToPDFFile(const AHTMLFileName, APDFFileName: string)	Crea un archivo PDF a partir de un archivo HTML.
RenderHTML(const AHTML: string; ADocument: TsgcPDFDocument; const ABasePath: string = ''): Integer	Agrega las páginas a un documento y devuelve el número de páginas agregadas. RenderHTMLW toma un WideString, RenderHTMLBytes bytes en bruto (la codificación proviene del BOM, <meta charset> o detección de UTF-8) y RenderHTMLFile un archivo.

ABasePath resuelve las rutas relativas de imágenes y hojas de estilo. En versiones de Delphi anteriores a 2009, un AnsiString puede contener UTF-8 o Windows-1252, lo cual se detecta.

Propiedades

Propiedad	Descripción
PageWidth, PageHeight: Double	Tamaño de página en puntos. El @page size de CSS lo sobrescribe.
Margins: TsgcPDFHTMLMargins	Left, Top, Right y Bottom en puntos. El @page margin de CSS los sobrescribe.
DefaultFontName, DefaultFontSize	Fuente del texto sin font-family ni font-size.
TrueTypeFonts: Boolean	Incrusta familias de fuentes distintas de las 14 estándar como fuentes TrueType. Necesario para texto fuera del conjunto de caracteres Windows-1252.
HeaderText, FooterText: string	Texto centrado en los márgenes superior e inferior. {page} y {pages} se reemplazan por el número de página y el número total de páginas.
MaxPages: Integer	Se detiene después de este número de páginas.
OnPageHeader, OnPageFooter(Sender; ACanvas: TsgcPDFCanvas; APageNumber, APageCount: Integer)	Dibuje su propio encabezado y pie de página con un TsgcPDFCanvas.
LastPageCount	Páginas agregadas por el último renderizado.

HTML admitido

- Elementos de bloque: html, body, div, p, h1 a h6, blockquote, pre, hr, address, center, section, article, header, footer, nav, aside, main, figure, figcaption, form, fieldset, details, summary.
- Listas: ul, ol y li (disc, circle, square, decimal, decimal-leading-zero, alfa en minúsculas y mayúsculas, latino y romano), dl, dt y dd.
- Elementos en línea: span, a, b, strong, i, em, u, ins, s, strike, del, small, big, sub, sup, code, kbd, samp, tt, cite, var, dfn, br.
- Tablas: table, caption, thead, tbody, tfoot, tr, th y td, con colspan y rowspan. Las filas de encabezado se repiten en todas las páginas.
- Imágenes: img con archivos JPEG, PNG, BMP y TIFF (rutas locales relativas a ABasePath) y URI data: en base64, con width y height.
- Enlaces: a href se convierte en una anotación de enlace.
- Hojas de estilo: bloques <style>, atributos style y <link rel="stylesheet"> hacia archivos locales.

CSS admitido

- Selectores: elemento, clase, id, combinadores de descendiente y de hijo, :first-child, con especificidad, herencia e !important.
- Modelo de caja: margin, padding, border (width, style solid, dashed y dotted, color, por lado), width, max-width, height, background-color, border-collapse y border-spacing.
- Texto: color, font, font-family, font-size (incluidas palabras clave, em, rem, %, px, pt, mm, cm, in), font-weight, font-style, line-height, letter-spacing, text-align (incluido justify), text-indent, text-decoration, vertical-align, white-space.
- Listas y display: list-style, list-style-type, display (block, inline, list-item, table y partes de table, none).
- Paginación: page-break-before, page-break-after, page-break-inside, break-before, break-after, break-inside, y @page con size (A3, A4, A5, B5, letter, legal, landscape) y margin. Las líneas, imágenes y filas de tabla nunca se dividen entre páginas.
- Colores como nombres, #rgb, #rrggbb, rgb() y rgba().

No admitido: JavaScript, floats, posicionamiento absolute y fixed, flexbox, grid, fuentes web (@font-face), imágenes cargadas por HTTP, SVG y controles de formulario. Las propiedades no admitidas se ignoran.

Ejemplo: cadena HTML a PDF

```
procedure HTMLToPDF(const APDFFile: string);
var
  oHTML: TsgcPDFHTMLRenderer;
begin
  oHTML := TsgcPDFHTMLRenderer.Create(nil);
  try
    oHTML.PageWidth := 595;
    oHTML.PageHeight := 842;
    oHTML.FooterText := 'Page {page} of {pages}';
    oHTML.HTMLToPDFFile(
      '<html><head><style>' +
      'h1 { color: #003366; border-bottom: 1px solid #999 }' +
      'td, th { border: 1px solid #ccc; padding: 4px }' +
      '.total { font-weight: bold; text-align: right }' +
      '</style></head><body>' +
      '<h1>Invoice 2026-001</h1>' +
      '<p>Thank you for your order. See <a href="https://www.esegece.com">' +
      'our website</a>.</p>' +
      '<table><tr><th>Item</th><th>Amount</th></tr>' +
      '<tr><td>sgcPDF license</td><td class="total">299.00</td></tr></table>' +
      '<ul><li>Delivery by download</li><li>One year of updates</li></ul>' +
      '</body></html>', APDFFile);
  finally
    oHTML.Free;
  end;
end;
```

Ejemplo: agregar HTML a un documento

```
function AppendHTML(ADocument: TsgcPDFDocument;
  const AHTMLFile: string): Integer;
var
  oHTML: TsgcPDFHTMLRenderer;
begin
  oHTML := TsgcPDFHTMLRenderer.Create(nil);
  try
    oHTML.Margins.Left := 50;
    oHTML.Margins.Right := 50;
    oHTML.DefaultFontName := 'Arial';
    oHTML.DefaultFontSize := 11;
    // relative image and stylesheet paths resolve from the HTML file folder
    Result := oHTML.RenderHTMLFile(AHTMLFile, ADocument);
  finally
    oHTML.Free;
  end;
end;
```

Vea también

- TsgcPDFCanvas
- Tablas

Códigos de barras 2D y GS1-128

TsgcPDFBarcodeGenerator (unidad sgcPDF_Barcodes) dibuja símbolos 2D y códigos de barras GS1 como gráficos vectoriales, además de los códigos de barras lineales descritos en Códigos de barras. El parámetro APage de cada método Draw acepta un TsgcPDFPage o su generador de contenido (`Page.Content`). Cualquier otro objeto genera una excepción.

Simbología	Método	Opciones
QR Code	DrawQRCode(APage; const AData: AnsiString; AX, AY, AModuleSize: Double; AErrorLevel: TsgcPDFQRErrorLevel; AMode: TsgcPDFQRMode; AVersion: Integer)	Versiones 1 a 40 (0 elige la más pequeña que se ajuste; una versión fija genera una excepción cuando los datos no caben), corrección de errores qrecL, qrecM, qrecQ, qrecH, modos qrmAuto, qrmNumeric, qrmAlphanumeric, qrmByte. Las sobrecargas más cortas utilizan los valores predeterminados.
DataMatrix ECC200	DrawDataMatrix(APage; const AData: AnsiString; AX, AY, AModuleSize: Double; AShape: TsgcPDFDataMatrixShape = dmsSquare)	24 tamaños cuadrados y 6 rectangulares, dmsSquare, dmsRectangle o dmsAuto (el símbolo más pequeño de cualquier forma). Codificación ASCII y Base256.
PDF417	DrawPDF417(APage; const AData: AnsiString; AX, AY, AModuleWidth: Double; ASecurityLevel: Integer = -1; AColumns: Integer = 0)	Niveles de seguridad de 0 a 8 (-1 automático), de 1 a 30 columnas de datos (0 automático), de 3 a 90 filas. Compactación de texto, numérica y de bytes.
GS1-128	DrawBarcode(APage; bcGS1_128; const AData: AnsiString; AX, AY: Double [; const AOptions])	Identificadores de aplicación entre paréntesis, por ejemplo (01)09501101530003(17)261231. Se inserta el FNC1 y se verifica el dígito de control del GTIN.

DataMatrix y PDF417 también están disponibles mediante DrawBarcode con bcDataMatrix y bcPDF417, utilizando los campos QRVersion, QRErrorLevel, QRMode, DataMatrixShape, PDF417SecurityLevel, PDF417Columns y PDF417RowHeight de TsgcPDFBarcodeOptions. Los codificadores `EncodeQRCodeMatrix`, `EncodeDataMatrix`, `EncodePDF417` y `EncodeGS1_128`, y `EncodeLinear` para cualquier simbología lineal, devuelven la matriz de módulos o los anchos de barra si desea dibujar el símbolo usted mismo. TsgcPDFBarcodeSymbol hace ese trabajo para cualquier simbología: añade la zona de silencio, la línea de texto y el giro, y es el símbolo que muestran los controles de código de barras de VCL y FireMonkey.

Ejemplo

```
procedure Create2DBarcodes(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  oBarcodes: TsgcPDFBarcodeGenerator;
  oOptions: TsgcPDFBarcodeOptions;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPage := oPDF.AddPage(595, 842);
    oBarcodes := TsgcPDFBarcodeGenerator.Create;
    try
      // QR Code: level H, alphanumeric mode, version 0 = smallest that fits
      oBarcodes.DrawQRCode(oPage, 'HTTPS://WWW.ESEGECE.COM', 50, 650, 3,
        qrecH, qrmAlphanumeric, 0);
      // DataMatrix ECC200, square or rectangular symbols
      oBarcodes.DrawDataMatrix(oPage, 'SGC-PDF-2026', 250, 700, 3);
      oBarcodes.DrawDataMatrix(oPage, 'SGC-PDF-2026', 350, 700, 3,
```

```
        dmsRectangle);
      // PDF417: security level 5, 4 data columns
      oBarcodes.DrawPDF417(oPage, 'sgcPDF PDF417 sample text', 50, 520, 1, 5, 4);
      // GS1-128: application identifiers in parentheses
      oOptions := sgcPDFDefaultBarcodeOptions;
      oOptions.Height := 40;
      oBarcodes.DrawBarcode(oPage, bcGS1_128, '(01)09501101530003(17)261231',
        50, 400, oOptions);
    finally
      oBarcodes.Free;
    end;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

No implementado: codificación C40 y Text de DataMatrix, y segmentación de modo mixto de QR Code. La salida sigue siendo válida, solo menos compacta para algunos datos.

Vea también

- Códigos de barras
- TsgcPDFCanvas
- TsgcPDFBarcodeSymbol
- TsgcPDFBarcodeControl

TsgcPDFBarcodeControl

TsgcPDFBarcodeControl (unidad sgcPDF_Barcode_VCL) es un control VCL que muestra un código de barras o un símbolo 2D en un formulario. Aloja un TsgcPDFBarcodeSymbol, que codifica el texto con los codificadores de TsgcPDFBarcodeGenerator, y dibuja un rectángulo relleno por cada serie de módulos oscuros, de modo que cada módulo empieza y termina en un píxel entero y el símbolo se mantiene nítido para los lectores. Admite las once simbologías de sgcPDF: Code 128, Code 39, EAN-13, EAN-8, UPC-A, ITF, Codabar, GS1-128, QR Code, DataMatrix y PDF417.
El control dibuja solo con el canvas de la VCL, sin GDI+, por lo que funciona desde Delphi 7. Lo instalan los paquetes VCL, en la página de paleta **sgcPDF**. El mismo símbolo se puede guardar como PNG o SVG, o dibujar en una página PDF con los mismos módulos que en pantalla.

Propiedades publicadas

Propiedad	Descripción
Symbology: TsgcPDFBarcodeType	La simbología: bcCode128 (por defecto), bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcGS1_128, o las simbologías 2D bcQRCode, bcDataMatrix y bcPDF417.
Text: string	Los datos que se codifican. QR Code, DataMatrix y PDF417 codifican los bytes UTF-8 del texto. Una simbología lineal solo acepta su propio juego de caracteres, y GS1-128 recibe los identificadores de aplicación entre paréntesis.
QuietZone: Integer	El margen claro en módulos. -1 (por defecto) usa el estándar de la simbología: 10 para los símbolos lineales, a izquierda y derecha, y 4 para QR Code, 1 para DataMatrix y 2 para PDF417, en los cuatro lados.
BarHeight: Integer	La altura de las barras de un símbolo lineal en módulos. 0 (por defecto) rellena la altura disponible.
ShowText: Boolean	Dibuja la línea legible bajo las barras de un símbolo lineal. Por defecto True. Los símbolos 2D no tienen línea de texto.
Rotation: TsgcPDFBarcodeRotation	br0 (por defecto), br90, br180 o br270: giros de un cuarto de vuelta en sentido horario. La línea de texto gira con el símbolo.
Fit: TsgcPDFBarcodeFit	bfFitInteger (por defecto): el mayor número entero de píxeles por módulo que cabe, centrado. bfModuleSize: exactamente ModuleSize píxeles por módulo, centrado. bfStretch: el símbolo llena el control y los módulos pueden diferir en un píxel.
ModuleSize: Integer	Píxeles por módulo para bfModuleSize. Por defecto 2. Se escala con el formulario en pantallas de alta densidad.
Color: TColor	El fondo y los módulos claros. Por defecto clWhite.
BarColor: TColor	Los módulos oscuros. Por defecto clBlack.
Font, ParentFont	La fuente, el estilo y el color de la línea de texto. Su altura sigue el tamaño del módulo (8 módulos en una banda de 12 módulos bajo las barras), por lo que Font.Size se ignora. Siempre se usa una fuente TrueType.
Transparent: Boolean	Deja el fondo sin pintar, de modo que el padre se ve entre las barras. Por defecto False.

Propiedad	Descripción
QRErrorLevel: TsgcPDFQRErrorLevel	Corrección de errores de QR Code: qrecL, qrecM (por defecto), qrecQ o qrecH.
QRMode: TsgcPDFQRMode	Modo de codificación de QR Code: qrmByte (por defecto), qrmAuto, qrmNumeric o qrmAlphanumeric.
QRVersion: Integer	0 (por defecto) elige la versión de QR Code más pequeña que cabe, de 1 a 40 fija la versión.
DataMatrixShape: TsgcPDFDataMatrixShape	dmsSquare (por defecto), dmsRectangle o dmsAuto (el símbolo más pequeño de cualquier forma).
PDF417SecurityLevel: Integer	Nivel de seguridad de PDF417 de 0 a 8, -1 (por defecto) es automático.
PDF417Columns: Integer	Columnas de datos de PDF417 de 1 a 30, 0 (por defecto) es automático.
PDF417RowHeight: Integer	La altura de una fila de PDF417 en módulos. Por defecto 3.
Align, Anchors, Visible, Hint	Propiedades estándar de la VCL.

Eventos

Evento	Descripción
OnError: TsgcPDFBarcodeErrorEvent	Se llama con Sender y aError (Symbol.LastError) cuando el texto no se puede codificar, una vez por cada cambio del símbolo. El control dibuja entonces un marco fino rayado. Dibujar nunca lanza una excepción.
OnClick: TNotifyEvent	Se hizo clic en el control.

Métodos públicos

Método	Descripción
IsValid: Boolean	True cuando el texto se puede codificar. Symbol.LastError indica por qué no.
PaintTo(aCanvas: TCanvas; const aRect: TRect)	Dibuja el símbolo en aRect de cualquier canvas, por ejemplo un mapa de bits o el canvas de una impresora, como se dibuja el propio control: el fondo (salvo con Transparent), los módulos y la línea de texto, o el marco rayado cuando el texto no es válido.
SaveToPNG(const aFileName: string; aModuleSize: Integer), SaveToPNG(aStream: TStream; aModuleSize: Integer)	Guarda el símbolo como imagen PNG con aModuleSize píxeles por módulo, BarColor sobre Color. La línea de texto usa la fuente de Font y BarColor.
SaveToSVG(const aFileName: string; aModuleSize: Integer), SaveToSVG(aStream: TStream; aModuleSize: Integer)	Guarda el símbolo como SVG con aModuleSize unidades por módulo, con los mismos colores y la misma fuente que SaveToPNG.

Método	Descripción
Symbol: TsgcPDFBarcodeSymbol	El símbolo alojado, de solo lectura. El control es su propietario y usa su OnChange. Úselo para dibujar el símbolo en una página PDF (DrawToPDF) o para leer Runs, ModuleWidth y ModuleHeight.

SaveToPNG y SaveToSVG lanzan una excepción con Symbol.LastError cuando el texto no es válido, antes de crear ningún archivo.

Ejemplo

```
procedure TMainForm.CreateBarcode;
begin
  FBarcode := TsgcPDFBarcodeControl.Create(Self);
  FBarcode.Parent := Self;
  FBarcode.SetBounds(16, 16, 320, 140);
  FBarcode.Symbology := bcEAN13;
  FBarcode.Text := '5901234123457';
  FBarcode.BarHeight := 60;
  FBarcode.BarColor := clNavy;
  FBarcode.Font.Name := 'Arial';
  FBarcode.OnError := OnBarcodeError;
end;

procedure TMainForm.OnBarcodeError(Sender: TObject; const aError: string);
begin
  // the control draws a hatched frame, the event tells why
  Caption := aError;
end;

procedure TMainForm.SaveBarcode(const aFileName: string);
begin
  if FBarcode.IsValid then
  begin
    // 4 pixels per module, BarColor on Color
    FBarcode.SaveToPNG(ChangeFileExt(aFileName, '.png'), 4);
    FBarcode.SaveToSVG(ChangeFileExt(aFileName, '.svg'), 4);
  end;
end;
```

Añadir el control a un PDF

TsgcPDFBarcodeSymbol.DrawToPDF dibuja el símbolo del control en una página PDF como rectángulos vectoriales, con los mismos módulos, giro, zona de silencio y línea de texto que en pantalla. DrawToPDF recibe los colores como $RRGGBB, mientras que un TColor de la VCL es $BBGGRR, así que intercambie primero los bytes de rojo y azul.

```
procedure AddBarcodeToPDF(aControl: TsgcPDFBarcodeControl;
  const aFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  vFore, vBack: Cardinal;
begin
  // TColor is $BBGGRR and DrawToPDF takes $RRGGBB: swap red and blue
  vFore := Cardinal(ColorToRGB(aControl.BarColor));
  vFore := ((vFore and $FF) shl 16) or (vFore and $FF00) or
    ((vFore shr 16) and $FF);
  vBack := Cardinal(ColorToRGB(aControl.Color));
  vBack := ((vBack and $FF) shl 16) or (vBack and $FF00) or
    ((vBack shr 16) and $FF);
  oPDF := TsgcPDFDocument.Create;
  try
```

```
    oPage := oPDF.AddPage(595, 842);
    // the modules, rotation, quiet zone and text line of the control,
    // fitted in a 250 x 100 points box whose bottom left corner is 50, 680
    aControl.Symbol.DrawToPDF(oPage, 50, 680, 250, 100, vFore, vBack,
      sfHelvetica);
    oPDF.SaveToFile(aFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFBarcodeFMX
- TsgcPDFBarcodeSymbol
- Códigos de barras
- Códigos de barras 2D y GS1-128

TsgcPDFBarcodeFMX

TsgcPDFBarcodeFMX (unidad sgcPDF_Barcode_FMX) es la versión FireMonkey de TsgcPDFBarcodeControl. Aloja el mismo TsgcPDFBarcodeSymbol y dibuja un rectángulo por cada serie de módulos oscuros. El diseño se calcula en píxeles del dispositivo, con la escala del canvas, de modo que cada módulo empieza y termina en un píxel entero del dispositivo y se mantiene nítido en pantallas de alta densidad. El control recorta su dibujo a sus límites.
Requiere Delphi XE5 o posterior (en compiladores anteriores la unidad está vacía) y lo instalan los paquetes FMX, en la página de paleta **sgcPDF**.

Propiedades publicadas

Propiedad	Descripción
Symbology: TsgcPDFBarcodeType	La simbología: bcCode128 (por defecto), bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcGS1_128, o las simbologías 2D bcQRCode, bcDataMatrix y bcPDF417.
Text: string	Los datos que se codifican. QR Code, DataMatrix y PDF417 codifican los bytes UTF-8 del texto. Una simbología lineal solo acepta su propio juego de caracteres, y GS1-128 recibe los identificadores de aplicación entre paréntesis.
QuietZone: Integer	El margen claro en módulos. -1 (por defecto) usa el estándar de la simbología: 10 para los símbolos lineales, a izquierda y derecha, y 4 para QR Code, 1 para DataMatrix y 2 para PDF417, en los cuatro lados.
BarHeight: Integer	La altura de las barras de un símbolo lineal en módulos. 0 (por defecto) rellena la altura disponible.
ShowText: Boolean	Dibuja la línea legible bajo las barras de un símbolo lineal. Por defecto True. Los símbolos 2D no tienen línea de texto.
Rotation: TsgcPDFBarcodeRotation	br0 (por defecto), br90, br180 o br270: giros de un cuarto de vuelta en sentido horario. La línea de texto gira con el símbolo.
Fit: TsgcPDFBarcodeFit	bfFitInteger (por defecto): el mayor número entero de píxeles del dispositivo por módulo que cabe, centrado. bfModuleSize: ModuleSize unidades lógicas por módulo, redondeadas a píxeles enteros del dispositivo, centrado. bfStretch: el símbolo llena el control y los módulos pueden diferir en un píxel del dispositivo.
ModuleSize: Integer	Unidades lógicas por módulo para bfModuleSize, como cualquier otro tamaño de FireMonkey. Por defecto 2.
Color: TAlphaColor	El fondo y los módulos claros. Por defecto blanco.
BarColor: TAlphaColor	Los módulos oscuros. Por defecto negro.
TextSettings: TsgcPDFBarcodeTextSettingsFMX	Font (familia y estilo) y FontColor de la línea de texto. Su altura sigue el tamaño del módulo, por lo que Font.Size se ignora.
Transparent: Boolean	Deja el fondo sin pintar. Por defecto False.
QRErrorLevel: TsgcPDFQRErrorLevel	Corrección de errores de QR Code: qrecL, qrecM (por defecto), qrecQ o qrecH.

Propiedad	Descripción
QRMode: TsgcPDFQRMode	Modo de codificación de QR Code: qrmByte (por defecto), qrmAuto, qrmNumeric o qrmAlphanumeric.
QRVersion: Integer	0 (por defecto) elige la versión de QR Code más pequeña que cabe, de 1 a 40 fija la versión.
DataMatrixShape: TsgcPDFDataMatrixShape	dmsSquare (por defecto), dmsRectangle o dmsAuto (el símbolo más pequeño de cualquier forma).
PDF417SecurityLevel: Integer	Nivel de seguridad de PDF417 de 0 a 8, -1 (por defecto) es automático.
PDF417Columns: Integer	Columnas de datos de PDF417 de 1 a 30, 0 (por defecto) es automático.
PDF417RowHeight: Integer	La altura de una fila de PDF417 en módulos. Por defecto 3.
Align, Anchors, Margins, Position, Width, Height, Size, Visible, Hint, ShowHint	Propiedades estándar de FireMonkey. Size se publica desde Delphi XE7, y Hint y ShowHint desde Delphi XE6.

Eventos

Evento	Descripción
OnError: TsgcPDFBarcodeErrorEvent	Se llama con Sender y aError (Symbol.LastError) cuando el texto no se puede codificar, una vez por cada cambio del símbolo. El control dibuja entonces un marco fino rayado. Dibujar nunca lanza una excepción.
OnClick: TNotifyEvent	Se hizo clic o se tocó el control.

Métodos públicos

Método	Descripción
IsValid: Boolean	True cuando el texto se puede codificar. Symbol.LastError indica por qué no.
SaveToPNG(const aFileName: string; aModuleSize: Integer), SaveToPNG(aStream: TStream; aModuleSize: Integer)	Guarda el símbolo como imagen PNG con aModuleSize píxeles por módulo, BarColor sobre Color. La línea de texto usa la familia de TextSettings.Font.
SaveToSVG(const aFileName: string; aModuleSize: Integer), SaveToSVG(aStream: TStream; aModuleSize: Integer)	Guarda el símbolo como SVG con aModuleSize unidades por módulo, con los mismos colores y la misma fuente que SaveToPNG.
Symbol: TsgcPDFBarcodeSymbol	El símbolo alojado, de solo lectura. El control es su propietario y usa su OnChange.

SaveToPNG y SaveToSVG lanzan una excepción con Symbol.LastError cuando el texto no es válido, antes de crear ningún archivo.

Colores

Un TAlphaColor es $AARRGGBB, y los métodos de TsgcPDFBarcodeSymbol reciben $RRGGBB e ignoran el byte superior. Así, Color y BarColor se pasan tal cual a DrawToPDF, SaveToSVG y SaveToPNG del símbolo, sin intercambio de bytes.

Ejemplo

```
procedure TMainForm.CreateQRCode;
begin
  FQRCode := TsgcPDFBarcodeFMX.Create(Self);
  FQRCode.Parent := Self;
  FQRCode.Align := TAlignLayout.Client;
  FQRCode.Symbology := bcQRCode;
  FQRCode.QRErrorLevel := qrecH;
  FQRCode.Text := 'https://www.esegece.com';
  FQRCode.BarColor := TAlphaColors.Navy;
  FQRCode.OnError := OnBarcodeError;
end;

procedure TMainForm.OnBarcodeError(Sender: TObject; const aError: string);
begin
  Caption := aError;
end;

procedure TMainForm.SaveQRCode;
begin
  // on mobile, write to the documents folder of the application
  if FQRCode.IsValid then
    FQRCode.SaveToPNG(TPath.Combine(TPath.GetDocumentsPath, 'qrcode.png'), 8);
end;

procedure TMainForm.AddQRCodeToPDF(const aFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // a TAlphaColor is $AARRGGBB and DrawToPDF ignores the top byte
    FQRCode.Symbol.DrawToPDF(oPDF.AddPage(595, 842), 50, 600, 200, 200,
      FQRCode.BarColor, FQRCode.Color);
    oPDF.SaveToFile(aFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- TsgcPDFBarcodeControl
- TsgcPDFBarcodeSymbol
- macOS, iOS y Android

TsgcPDFBarcodeSymbol

TsgcPDFBarcodeSymbol (unidad sgcPDF_Barcode_Symbol) es el código de barras independiente del framework que hay detrás de TsgcPDFBarcodeControl y TsgcPDFBarcodeFMX. Codifica un código de barras con los codificadores de TsgcPDFBarcodeGenerator, lo dispone en una cuadrícula de módulos y lo escribe como texto SVG, como imagen PNG o como rectángulos vectoriales en una página PDF. No usa VCL, FMX ni GDI, por lo que forma parte del paquete core y también funciona en servicios, aplicaciones de consola y en Linux de 64 bits.

La cuadrícula de módulos

- La cuadrícula contiene todo el símbolo después del giro, incluida la zona de silencio y, en una simbología lineal con ShowText, la banda de la línea legible.
- La banda de texto mide 12 módulos de alto y queda bajo las barras antes del giro. El texto mide 8 módulos de alto, su línea base está a 10 módulos del borde de la banda junto a las barras y está centrado a lo largo de la banda. Los caracteres de control se muestran como espacios.
- Un símbolo lineal con BarHeight 0 cuenta 50 módulos de barras en ModuleHeight y en la salida SVG y PNG. Los controles y DrawToPDF hacen en cambio que sus barras llenen la altura disponible.
- Rotation gira el símbolo en sentido horario en cuartos de vuelta.
- Los colores son $RRGGBB y el byte superior se ignora, así que un TAlphaColor de FireMonkey se pasa tal cual, mientras que los bytes de rojo y azul de un TColor de la VCL deben intercambiarse.

Propiedades publicadas

Propiedad	Descripción
Symbology: TsgcPDFBarcodeType	La simbología: bcCode128 (por defecto), bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcGS1_128, o las simbologías 2D bcQRCode, bcDataMatrix y bcPDF417.
Text: string	Los datos que se codifican. QR Code, DataMatrix y PDF417 codifican los bytes UTF-8 del texto. Una simbología lineal solo acepta su propio juego de caracteres, y GS1-128 recibe los identificadores de aplicación entre paréntesis.
QuietZone: Integer	El margen claro en módulos. -1 (por defecto) usa el estándar de la simbología: 10 para los símbolos lineales, a izquierda y derecha, y 4 para QR Code, 1 para DataMatrix y 2 para PDF417, en los cuatro lados.
BarHeight: Integer	La altura de las barras de un símbolo lineal en módulos. 0 (por defecto) rellena la altura disponible.
ShowText: Boolean	Dibuja la línea legible bajo las barras de un símbolo lineal. Por defecto True. Los símbolos 2D no tienen línea de texto.
Rotation: TsgcPDFBarcodeRotation	br0 (por defecto), br90, br180 o br270: giros de un cuarto de vuelta en sentido horario. La línea de texto gira con el símbolo.
QRErrorLevel: TsgcPDFQRErrorLevel	Corrección de errores de QR Code: qrecL, qrecM (por defecto), qrecQ o qrecH.
QRMode: TsgcPDFQRMode	Modo de codificación de QR Code: qrmByte (por defecto), qrmAuto, qrmNumeric o qrmAlphanumeric.
QRVersion: Integer	0 (por defecto) elige la versión de QR Code más pequeña que cabe, de 1 a 40 fija la versión.

Propiedad	Descripción
DataMatrixShape: TsgcPDFDataMatrixShape	dmsSquare (por defecto), dmsRectangle o dmsAuto (el símbolo más pequeño de cualquier forma).
PDF417SecurityLevel: Integer	Nivel de seguridad de PDF417 de 0 a 8, -1 (por defecto) es automático.
PDF417Columns: Integer	Columnas de datos de PDF417 de 1 a 30, 0 (por defecto) es automático.
PDF417RowHeight: Integer	La altura de una fila de PDF417 en módulos. Por defecto 3.
OnChange: TNotifyEvent	Cambió una propiedad. Los controles de código de barras usan el OnChange de su Symbol para repintar, así que no lo reemplace allí.

Métodos y propiedades públicos

Miembro	Descripción
Encode: Boolean	Codifica el símbolo cuando una propiedad cambió desde la última llamada y devuelve True cuando el texto es válido. Los demás métodos lo llaman cuando hace falta.
LastError: string	Por qué no se puede codificar el texto.
ModuleWidth, ModuleHeight: Integer	El tamaño de la cuadrícula girada en módulos, con la zona de silencio y la banda de texto incluidas. 0 cuando el texto no es válido.
ModuleAt(aCol, aRow: Integer): Boolean	True cuando el módulo en aCol, aRow de la cuadrícula girada es oscuro.
Runs: TsgcPDFBarcodeRuns	Los módulos oscuros como rectángulos TsgcPDFBarcodeRun (Col, Row, Cols, Rows), combinados por fila y luego entre filas, ordenados por fila y columna. Vacío cuando el texto no es válido.
TextBounds(out aCol, aRow, aCols, aRows: Integer): Boolean	La banda de la línea de texto en la cuadrícula girada. False cuando el símbolo no dibuja texto: simbologías 2D, ShowText desactivado o un texto no válido.
FitModuleSize(aWidth, aHeight: Integer): Integer	El mayor tamaño entero de módulo con el que el símbolo cabe en aWidth por aHeight píxeles, 0 cuando un píxel por módulo no cabe. En un símbolo lineal con BarHeight 0 solo cuentan, a lo ancho de las barras, la banda de texto y un módulo de barras, porque las barras llenan la altura.
LayoutAxes(aFit: TsgcPDFBarcodeFit; aModuleSize, aLeft, aTop, aWidth, aHeight: Integer; out aX, aY: TsgcPDFBarcodeAxis)	La disposición completa en píxeles de la cuadrícula dentro de un cuadro, la que usan los controles para dibujar. Véase Dibujo personalizado.
SaveToSVG(const aFileName: string; aModuleSize: Integer; aFore, aBack: Cardinal; const aFontName: string), SaveToSVG(aStream: TStream; ...)	Escribe el símbolo como SVG con aModuleSize unidades por módulo, aFore sobre aBack. La línea de texto usa la familia de fuentes aFontName.
SaveToPNG(const aFileName: string; aModuleSize: Integer; aFore, aBack: Cardinal; const aFontName: string), SaveToPNG(aStream: TStream; ...)	Escribe el símbolo como imagen PNG con aModuleSize píxeles por módulo, mediante el canvas raster en Pascal puro. La línea de texto usa la fuente TrueType del sistema operativo aFontName, o una

Miembro	Descripción
	fuente sans serif si no está instalada. Sin ninguna fuente utilizable, el PNG no tiene línea de texto.
DrawToPDF(aPage: TObject; aX, aY, aWidth, aHeight: Double; aFore, aBack: Cardinal; aFont: TsgcPDFStandardFont = sfHelvetica)	Dibuja el símbolo en una página PDF. Véase Dibujar en una página PDF.
Assign(Source: TPersistent)	Copia las propiedades publicadas de otro símbolo.

SaveToSVG y SaveToPNG lanzan una excepción cuando aModuleSize es menor que 1, y con LastError cuando el texto no es válido, antes de crear ningún archivo.

Dibujar en una página PDF

DrawToPDF dibuja el símbolo como gráficos vectoriales, de modo que una página PDF muestra exactamente lo que muestra un control de código de barras. Sustituye la asignación manual de las propiedades de un control a TsgcPDFBarcodeOptions: el giro, la zona de silencio, la altura de las barras y la línea de texto proceden del símbolo. TsgcPDFBarcodeGenerator.DrawBarcode sigue siendo la forma más corta de dibujar un código de barras directamente a partir de datos.

Parámetro	Descripción
aPage: TObject	Un TsgcPDFPage o su generador de contenido, `TsgcPDFPage.Content`, como en DrawBarcode. Cualquier otro objeto lanza una excepción.
aX, aY: Double	La esquina inferior izquierda del cuadro en puntos. El eje y del PDF apunta hacia arriba.
aWidth, aHeight: Double	El tamaño del cuadro en puntos. Un cuadro vacío lanza una excepción.
aFore: Cardinal	El color de las barras y de la línea de texto, $RRGGBB.
aBack: Cardinal	El color pintado bajo todo el símbolo, zona de silencio incluida, $RRGGBB.
aFont: TsgcPDFStandardFont	La fuente estándar de PDF de la línea de texto. Por defecto sfHelvetica.

- Tamaño: el símbolo toma el mayor tamaño de módulo que cabe en el cuadro, no solo en puntos enteros, y queda centrado en él. Un símbolo lineal con BarHeight 0 llena el cuadro a lo ancho de las barras, como FitModuleSize y LayoutAxes.
- Salida: el rectángulo aBack, luego un rectángulo relleno por cada entrada de Runs en aFore, de modo que cada borde de módulo cae exactamente sobre la cuadrícula, y después la línea de texto, todo dentro de un estado gráfico guardado.
- Texto: la línea de texto se dibuja en aFont con un tamaño de 8 módulos, su línea base a 10 módulos del borde de la banda junto a las barras, centrada en la banda, y gira en sentido horario con el símbolo.
- Errores: un texto no válido lanza una excepción con LastError y no dibuja nada.

Ejemplo

Una etiqueta GS1-128 dibujada dos veces en una página PDF, la segunda girada un cuarto de vuelta, y guardada como SVG y PNG, sin formulario y sin control:

```
procedure CreateShippingLabel(const aFileName: string);
var
  oSymbol: TsgcPDFBarcodeSymbol;
  oPDF: TsgcPDFDocument;
```

```
  oPage: TsgcPDFPage;
begin
  oSymbol := TsgcPDFBarcodeSymbol.Create;
  try
    oSymbol.Symbology := bcGS1_128;
    oSymbol.Text := '(01)09501101530003(17)261231';
    oSymbol.BarHeight := 30;
    if not oSymbol.Encode then
      raise Exception.Create(oSymbol.LastError);
    oPDF := TsgcPDFDocument.Create;
    try
      oPage := oPDF.AddPage(595, 842);
      // black on white in a 300 x 90 points box, bottom left corner 50, 700
      oSymbol.DrawToPDF(oPage, 50, 700, 300, 90, $000000, $FFFFFF);
      // a quarter turn clockwise, navy, the text line in Courier
      oSymbol.Rotation := br90;
      oSymbol.DrawToPDF(oPage.Content, 450, 350, 90, 300, $000080, $FFFFFF,
        sfCourier);
      oPDF.SaveToFile(aFileName);
    finally
      oPDF.Free;
    end;
    // the same symbol as SVG (2 units per module) and PNG (3 pixels)
    oSymbol.Rotation := br0;
    oSymbol.SaveToSVG(ChangeFileExt(aFileName, '.svg'), 2, $000000, $FFFFFF,
      'Arial');
    oSymbol.SaveToPNG(ChangeFileExt(aFileName, '.png'), 3, $000000, $FFFFFF,
      'Arial');
  finally
    oSymbol.Free;
  end;
end;
```

Dibujo personalizado

LayoutAxes calcula la disposición completa en píxeles con la que dibujan los controles, para un cuadro de aWidth por aHeight píxeles cuya esquina superior izquierda es aLeft, aTop. aFit fija el tamaño del módulo (aModuleSize píxeles con bfModuleSize), el símbolo se centra y un símbolo lineal con BarHeight 0 llena el cuadro a lo ancho de las barras. Devuelve un TsgcPDFBarcodeAxis por eje, y `sgcPDFBarcodeAxisPos(aAxis, aIndex)` devuelve el píxel de la línea de cuadrícula aIndex (de 0 a Count). Las series vecinas comparten sus bordes, por lo que no aparecen huecos entre los módulos.

```
procedure PaintSymbol(aSymbol: TsgcPDFBarcodeSymbol; aCanvas: TCanvas;
  const aRect: TRect);
var
  vX, vY: TsgcPDFBarcodeAxis;
  vRuns: TsgcPDFBarcodeRuns;
  i: Integer;
begin
  if not aSymbol.Encode then
    Exit;
  // the largest whole module size that fits, centred in aRect
  aSymbol.LayoutAxes(bfFitInteger, 0, aRect.Left, aRect.Top,
    aRect.Right - aRect.Left, aRect.Bottom - aRect.Top, vX, vY);
  aCanvas.Brush.Style := bsSolid;
  aCanvas.Brush.Color := clBlack;
  vRuns := aSymbol.Runs;
  for i := 0 to Length(vRuns) - 1 do
    aCanvas.FillRect(Rect(sgcPDFBarcodeAxisPos(vX, vRuns[i].Col),
      sgcPDFBarcodeAxisPos(vY, vRuns[i].Row),
      sgcPDFBarcodeAxisPos(vX, vRuns[i].Col + vRuns[i].Cols),
      sgcPDFBarcodeAxisPos(vY, vRuns[i].Row + vRuns[i].Rows)));
end;
```

Tipos y constantes

Nombre	Descripción
TsgcPDFBarcodeRotation	br0, br90, br180, br270: giros de un cuarto de vuelta en sentido horario.
TsgcPDFBarcodeFit	bfFitInteger, bfModuleSize, bfStretch: cómo los controles y LayoutAxes dimensionan los módulos.
TsgcPDFBarcodeRun, TsgcPDFBarcodeRuns	Un rectángulo de módulos oscuros en la cuadrícula girada (Col, Row, Cols, Rows) y la matriz que devuelve Runs.
TsgcPDFBarcodeAxis	Cómo se asignan a píxeles enteros las líneas de cuadrícula de un eje: Origin, Module (píxeles por módulo) y Count (módulos); con Module 0 (bfStretch), Span píxeles se reparten entre Count módulos. FillStart, FillCount y FillPixels describen la banda que llena el cuadro, las barras de un símbolo lineal con BarHeight 0.
sgcPDFBarcodeAxisPos(const aAxis: TsgcPDFBarcodeAxis; aIndex: Integer): Integer	El píxel de la línea de cuadrícula aIndex en un eje de LayoutAxes.
TsgcPDFBarcodeErrorEvent	`procedure(Sender: TObject; const aError: string) of object` , el evento OnError de ambos controles de código de barras.
CS_SGC_PDF_BARCODE_TEXT_BAND, CS_SGC_PDF_BARCODE_TEXT_SIZE, CS_SGC_PDF_BARCODE_TEXT_BASELINE	La línea de texto en módulos: altura de banda 12, tamaño de fuente 8, línea base 10.
CS_SGC_PDF_BARCODE_BAR_HEIGHT	50, la altura de barras en módulos de un símbolo lineal con BarHeight 0 en ModuleHeight, SVG y PNG.

Vea también

- TsgcPDFBarcodeControl
- TsgcPDFBarcodeFMX
- Códigos de barras
- Códigos de barras 2D y GS1-128

Linux

El núcleo de sgcPDF compila y se ejecuta en Linux de 64 bits (Delphi 10.2 Tokyo y posterior con la plataforma Linux64). Esto cubre el trabajo típico de servidor: crear, editar, combinar, rellenar, cifrar y leer documentos PDF, y renderizar páginas a imágenes, por ejemplo en un servicio web o un trabajo por lotes.

Disponible en Linux

- TsgcPDFDocument, TsgcPDFEditor y el analizador y el escritor, incluidos los guardados incrementales, los object streams comprimidos y los archivos de más de 2 GB.
- Texto, fuentes TrueType y OpenType (cargadas desde archivos con AddTrueTypeFontFromFile, o desde las carpetas de fuentes del sistema), shaping, tablas, códigos de barras, anotaciones, marcadores, capas, formularios, adjuntos, PDF etiquetado, salida y validación de PDF/A.
- Cifrado con contraseña (RC4 y AES), redacción, el optimizador.
- Extracción de texto con posiciones de caracteres, TsgcPDFImageExtractor y todos los decodificadores de imagen.
- TsgcPDFRendererRaster: página a PNG o BMP, y píxeles de página.
- TsgcPDFCanvas y TsgcPDFHTMLRenderer.
- Firmas digitales mediante el evento OnExternalSign (por ejemplo con sgcSign o su propio HSM), vea Firmas digitales.

Solo Windows

- TsgcPDFRendererGDI, TsgcPDFViewer, TsgcPDFExporter y los controles VCL TsgcPDFViewerControl, TsgcPDFThumbnailView y TsgcPDFOutlineView.
- TsgcPDFEMFPlayer y TsgcPDFVCLCanvasHelper.
- La firma y verificación con certificados de archivos PFX o del almacén de certificados de Windows, y el cifrado con certificados (clave pública), que utilizan CryptoAPI de Windows. En Linux estas llamadas generan una excepción.
- Las clases CJK de bajo nivel de sgcPDF_Fonts_CJK (el propio texto CJK funciona mediante AddTrueTypeFontFromFile).

Fuentes en Linux

`AddTrueTypeFont` busca fuentes instaladas en /usr/share/fonts y /usr/local/share/fonts. El renderizador ráster también examina ~/.fonts y ~/.local/share/fonts, y otras carpetas agregadas con `sgcPDFRasterAddFontDirectory`. Los servidores mínimos y los contenedores a menudo no tienen ninguna fuente: instale un paquete como fonts-dejavu-core, fonts-liberation o fonts-noto, o distribuya sus propias fuentes con la aplicación y utilice `AddTrueTypeFontFromFile`. Los documentos con fuentes incrustadas se renderizan correctamente sin ninguna fuente instalada.

Ejemplo: miniaturas en el servidor

El mismo código que en Windows:

```
procedure RenderPageToPNG(const APDFFile, APNGFile: string; ADPI: Integer);
var
  oPDF: TsgcPDFDocument;
  oRenderer: TsgcPDFRendererRaster;
  oPage: TsgcPDFPage;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    oPage := oPDF.Pages[0];
    oRenderer := TsgcPDFRendererRaster.Create;
    try
```

```
      // page size in points (1/72 inch) to pixels
      oRenderer.RenderPageToPNG(oPDF.Parser, oPage.PageDict,
        Round(oPage.Width * ADPI / 72), Round(oPage.Height * ADPI / 72),
        APNGFile);
    finally
      oRenderer.Free;
    end;
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- macOS, iOS y Android
- TsgcPDFRendererRaster
- Instalación

macOS, iOS y Android

Desde Delphi XE5, sgcPDF se puede utilizar en aplicaciones FireMonkey para macOS, iOS y Android, y para Windows y Linux (FMXLinux). Están disponibles las mismas unidades no Windows que en Linux, además del visor FireMonkey.

Característica	Unidades	Notas
Crear, editar, leer y extraer	Unidades del núcleo de sgcPDF	Igual que en Linux. Cargue las fuentes desde archivos distribuidos con la aplicación (AddTrueTypeFontFromFile), porque las plataformas móviles no exponen sus fuentes del sistema como archivos de forma portable.
Control de visor	sgcPDF_ViewerFMX	TsgcPDFViewerFMX: desplazamiento, zoom con pellizco, ajustar al ancho o a la página, enlaces.
Imágenes de página	sgcPDF_Renderer_FMX, sgcPDF_Renderer_Raster	TsgcPDFRendererFMX renderiza en un TBitmap de FMX, TsgcPDFRendererRaster en PNG, BMP o píxeles.
Firmas	sgcPDF_Signatures	Únicamente mediante OnExternalSign (sin CryptoAPI).

Instalación

Instale los paquetes FMX (sgcPDFFMXDxx y dclsgcPDFFMXDxx) además de los paquetes del núcleo para obtener TsgcPDFViewerFMX en la paleta. Para los destinos móviles, el IDE compila las unidades de la carpeta Source dentro de su aplicación: agregue la carpeta Source de sgcPDF a la ruta de la biblioteca de cada plataforma (Android de 32 y 64 bits, iOS Device de 64 bits, macOS de 64 bits y ARM de 64 bits, Linux de 64 bits).

Distribución de archivos

Utilice el Deployment Manager para agregar archivos PDF y fuentes a la aplicación, y ábralos desde la carpeta de documentos, por ejemplo con `TPath.Combine(TPath.GetDocumentsPath, 'sample.pdf')` (unidad System.IOUtils).

Vea también

- TsgcPDFViewerFMX
- Linux
- Arquitectura del renderizador

C++Builder

sgcPDF admite C++Builder 2007 a C++Builder 13 con paquetes de tiempo de ejecución y de tiempo de diseño, en la plataforma Win32 para todas las versiones y en Win64 desde C++Builder XE6 (incluida la plataforma Win64 Modern de C++Builder 13).

Paquetes

La carpeta Packages contiene, para cada versión de C++Builder, el paquete de tiempo de ejecución del núcleo **sgcPDFCxx.cbproj**, el paquete de tiempo de ejecución de VCL **sgcPDFVCLCxx.cbproj** y el paquete de tiempo de diseño **dclsgcPDFCxx.cbproj**, por ejemplo sgcPDFC13.cbproj, sgcPDFVCLC13.cbproj y dclsgcPDFC13.cbproj. Construya los dos paquetes de tiempo de ejecución, y luego construya e instale el paquete de tiempo de diseño. La construcción genera el encabezado .hpp de cada unidad y los archivos de importación .lib o .a.

Usar sgcPDF desde C++

- Incluya el encabezado de cada unidad que utilice, por ejemplo sgcPDF_Document.hpp, sgcPDF_Fonts.hpp y sgcPDF_Renderer_Raster.hpp.
- Las clases y funciones tienen los mismos nombres que en Delphi, dentro del espacio de nombres de su unidad (por ejemplo `Sgcpdf_document::TsgcPDFDocument`). La mayoría de los encabezados también se incorporan al espacio de nombres global mediante las directivas using generadas.
- Cree los componentes con `new` y libérelos con `delete`. Los componentes que descienden de TComponent toman un propietario en el constructor (pase NULL para ninguno).
- Los parámetros WideString y AnsiString aceptan las clases de cadena de C++Builder del mismo nombre, TBytes es `System::DynamicArray<System::Byte>`.

Los encabezados anulan la definición de las macros de Windows que colisionan con nombres de método (DrawText, FindText, GetObject, GetCharWidth, CreateWindow, CertNameToStr), de modo que métodos como `TsgcPDFTextExtractor::FindText` se pueden llamar con normalidad.

Vea también

- Instalación

Archivos adjuntos

Un PDF puede llevar otros archivos dentro de sí: los datos de origen de un informe, una hoja de cálculo, el XML de una factura. **TsgcPDFDocument.Attachments** (clase TsgcPDFAttachments, unidad sgcPDF_Attachments) agrega, enumera, extrae y elimina estos archivos incrustados. Se escriben en el árbol de nombres `/EmbeddedFiles` del catálogo y, con su relación, en el array `/AF` del catálogo que exige PDF/A-3.

Agregar archivos

```
procedure CreateWithAttachments(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oFile: TsgcPDFAttachment;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.AddPage(595, 842).DrawText('Report with attachments', 50, 780,
      sfHelveticaBold, 18);
    // from bytes: name, data, MIME type, description, relationship
    oPDF.Attachments.AddFromBytes('data.csv', StrBytes('id;qty'#13#10'1;10'),
      'text/csv', 'Source data of the report', afrSource);
    // from a file on disk, the file name becomes the attachment name
    oFile := oPDF.Attachments.AddFromFile(Dir + 'notes.txt', 'text/plain',
      'Notes');
    oFile.Relationship := afrSupplement;
    // the viewer opens the attachments panel
    oPDF.Attachments.ShowAttachmentsPanel := True;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Método	Descripción
AddFromBytes(AName; AData; AMimeType; ADescription; ARelationship)	Incrusta AData bajo el nombre AName. Solo AName y AData son obligatorios.
AddFromStream(AName; AStream; ...)	Incrusta el resto de AStream.
AddFromFile(AFileName; AMimeType; ADescription; ARelationship)	Incrusta un archivo desde disco. El adjunto se nombra igual que el archivo.
Delete(AIndex), Clear, IndexOf(AName)	Elimina adjuntos y busca uno por nombre.

Los nombres deben ser únicos: agregar un segundo archivo con el mismo nombre genera una excepción. Cada adjunto almacena su tamaño, las fechas de creación y modificación y una suma de comprobación MD5, tal como recomienda la especificación.

Relationship (AFRelationship)

ARelationship, o la propiedad `Relationship` de TsgcPDFAttachment, indica a un lector cómo se relaciona el archivo con el documento (PDF 2.0 y PDF/A-3):

Valor	Significado
afrSource	La fuente original del contenido del documento, por ejemplo la hoja de cálculo con la que se creó un informe.
afrData	Datos utilizados para construir el documento, por ejemplo los valores de un gráfico.
afrAlternative	Una representación alternativa del contenido, por ejemplo el XML de una factura electrónica.
afrSupplement	Una representación complementaria, por ejemplo una copia legible por máquina.
afrEncrypted-Payload	Un documento de carga útil cifrado.
afrFormData	Datos asociados con los campos de formulario del documento.
afrSchema	Una definición de esquema de un objeto asociado.
afrUnspecified	Sin relación indicada (predeterminado).

Lectura y extracción

Los documentos cargados rellenan la lista Attachments automáticamente. Cada TsgcPDFAttachment expone Name, Description, MimeType, Relationship, CreationDate, ModDate y Size, y devuelve sus datos con `GetBytes`, `SaveToStream` o `SaveToFile`.

```
procedure ListAttachments(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    for i := 0 to oPDF.Attachments.Count - 1 do
      WriteLn(string(oPDF.Attachments[i].Name), ' ',
        oPDF.Attachments[i].Size, ' bytes ',
        string(oPDF.Attachments[i].MimeType));
    i := oPDF.Attachments.IndexOf('data.csv');
    if i >= 0 then
      oPDF.Attachments[i].SaveToFile(Dir + 'extracted.csv');
    // remove one attachment and save a copy without it
    oPDF.Attachments.Delete(oPDF.Attachments.IndexOf('notes.txt'));
    oPDF.SaveToFile(Dir + 'attachments_deleted.pdf');
  finally
    oPDF.Free;
  end;
end;
```

Archivos adjuntos y PDF/A

Nivel	Regla
PDF/A-1b	No se permiten archivos incrustados: agregar uno genera una excepción.
PDF/A-2b	Solo archivos que sean a su vez documentos PDF/A. Establezca `Attachments.PDFACompliantFiles` en True para confirmarlo antes de agregarlos; en caso contrario, la operación genera una excepción.

Nivel	Regla
PDF/ A-3b	Cualquier tipo de archivo, cada uno con su AFRelationship en el array /AF del catálogo.

Establezca `ShowAttachmentsPanel` en True para escribir `/PageMode /UseAttachments`, de modo que el visor abra su panel de adjuntos. Los adjuntos se conservan en los documentos cifrados y con SaveIncremental.

Vea también

- Facturas Factur-X y ZUGFeRD
- PDF/A
- TsgcPDFDocument

Facturas Factur-X y ZUGFeRD

Factur-X (Francia) y ZUGFeRD 2 (Alemania) son el mismo formato híbrido de factura electrónica: un documento PDF/A-3 que las personas pueden leer, con la factura estructurada (XML UN/CEFACT Cross Industry Invoice) incrustada para el software de contabilidad. XRechnung utiliza el mismo contenedor con su propio nombre de archivo.
sgcPDF construye el contenedor PDF. El XML de la factura en sí lo crea su aplicación (o su ERP) y se pasa como bytes a **Attachments.AddFacturX**:

```
procedure CreateFacturX(const AInvoiceXML, AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oXML: TMemoryStream;
  vXML: TBytes;
  vFont: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  oXML := TMemoryStream.Create;
  try
    // the invoice XML (CII) is created by your application
    oXML.LoadFromFile(AInvoiceXML);
    SetLength(vXML, oXML.Size);
    Move(oXML.Memory^, vXML[0], oXML.Size);
    oPDF.Info.Title := 'Invoice 2026-001';
    // PDF/A requires embedded fonts
    vFont := oPDF.AddTrueTypeFont('Arial', False, False);
    oPDF.AddPage(595, 842).DrawTextTT('Invoice 2026-001', 50, 780, vFont, 18);
    // embeds factur-x.xml with the relationship of the profile, adds the
    // Factur-X XMP extension schema and saves the document as PDF/A-3b
    oPDF.Attachments.AddFacturX(vXML, fxpEN16931);
    oPDF.SaveToFile(AFileName);
  finally
    oXML.Free;
    oPDF.Free;
  end;
end;
```

Qué hace AddFacturX

- Incrusta el XML como **factur-x.xml** (**xrechnung.xml** para fxpXRechnung, o el AFileName que pase) con tipo MIME text/xml.
- Establece la relación que exige el perfil: **Data** para MINIMUM y BASIC WL, que no son facturas completas, y **Alternative** para BASIC, EN 16931, EXTENDED y XRECHNUNG.
- Agrega el esquema de extensión XMP de Factur-X (fx:DocumentType, fx:DocumentFileName, fx:Version, fx:ConformanceLevel) a los metadatos.
- Cambia un documento sin nivel de PDF/A a **PDF/A-3b**. Un documento ya establecido en PDF/A-1b o PDF/A-2b genera una excepción, ya que Factur-X exige PDF/A-3.
- Reemplaza un XML de factura anterior: un documento contiene una sola factura.

Profile (TsgcPDFFacturXProfile)	ConformanceLevel	Relationship
fxpMinimum	MINIMUM	Data
fxpBasicWL	BASIC WL	Data
fxpBasic	BASIC	Alternative
fxpEN16931	EN 16931	Alternative
fxpExtended	EXTENDED	Alternative

Profile (TsgcPDFFacturXProfile)	ConformanceLevel	Relationship
fxpXRechnung	XRECHNUNG	Alternative

El parámetro opcional AVersion establece fx:Version (predeterminado 1.0). Las reglas de PDF/A se aplican a todo el documento, así que dibuje el texto con fuentes TrueType incrustadas (AddTrueTypeFont y DrawTextTT).
Los documentos cargados informan de la factura en `Attachments.FacturX`, `FacturXProfile` y `FacturXFileName`.
El ejemplo anterior se comprobó con veraPDF (PDF/A-3b: PASS) y la factura fue extraída sin cambios por la biblioteca Python factur-x.

Vea también

- Archivos adjuntos
- PDF/A

Guardado incremental y salida compacta

SaveIncremental

TsgcPDFDocument.SaveIncremental actualiza un PDF cargado añadiendo una nueva revisión: los bytes originales se escriben sin cambios, seguidos únicamente de los objetos nuevos y modificados, una sección de referencias cruzadas del mismo tipo que la original (tabla o stream) y un trailer con `/Prev`. Las entradas `/ID` y `/Encrypt` se conservan.
Utilícelo para editar documentos firmados sin romper sus firmas, para agregar anotaciones o valores de formulario a archivos grandes con rapidez, y para conservar el historial de un documento.

```
procedure AppendChanges(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(AFileName);
    oPDF.Info.Title := 'Reviewed';
    oPDF.Annotations.AddText(0, 300, 700, 320, 720, 'Checked by QA');
    // the original bytes are kept, only the changes are appended
    oPDF.SaveIncremental(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Hay dos sobrecargas: `SaveIncremental(AStream: TStream)` escribe el archivo actualizado completo en un stream, y `SaveIncremental(AFileName: string)` escribe un archivo, que puede ser el mismo archivo desde el que se cargó el documento. Un documento sin cambios produce los bytes originales.
Las ediciones de anotaciones se guardan de la misma manera. Una anotación modificada del archivo se vuelve a escribir con su propio número de objeto, y las anotaciones y respuestas nuevas se añaden en la nueva revisión, de modo que el rango de bytes de cada firma existente sigue siendo el mismo. Un validador acepta la revisión cuando la firma permite comentarios: una firma de certificación con permiso DocMDP P=3, o solo firmas de aprobación. Con P=1 o P=2 informa de un cambio no permitido. Vea Anotaciones.

Límites

SaveIncremental genera una excepción, y se requiere un guardado completo (SaveToFile o SaveToStream), cuando:

- el documento no se cargó desde un PDF (es un documento nuevo);
- se agregaron otros documentos con AppendDocument;
- se agregaron nuevas capas o nuevos campos de formulario (rellenar campos existentes no supone un problema);
- se cambió el cifrado;
- el documento se redactó después de abrirse con una contraseña (vea TsgcPDFRedactor).

CompressObjects

Con **CompressObjects** establecido en True, un guardado completo escribe los objetos que no son streams en object streams comprimidos y las referencias cruzadas como un cross-reference stream comprimido (PDF 1.5). Los documentos con muchos objetos pequeños (páginas, anotaciones, campos de formulario, árboles de estructura) resultan entre un 39 y un 57 por ciento más pequeños en las pruebas. Se admite el cifrado. La propiedad se ignora para PDF/A-1, que no permite object streams.

```
procedure SaveCompact(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  i: Integer;
```

```
begin
  oPDF := TsgcPDFDocument.Create;
  try
    for i := 1 to 50 do
      oPDF.AddPage(595, 842).DrawText(AnsiString('Page ' + IntToStr(i)), 50,
        780, sfHelvetica, 12);
    // object streams and a compressed cross-reference stream (PDF 1.5)
    oPDF.CompressObjects := True;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Vea también

- Archivos grandes
- Sellos de tiempo y validación a largo plazo
- Optimización de archivos PDF
- Anotaciones

Archivos grandes

sgcPDF utiliza desplazamientos de archivo de 64 bits en todas partes, por lo que lee y escribe archivos de más de 2 GB, y no necesita cargar un archivo grande en memoria para trabajar con él.

MemoryThreshold

TsgcPDFDocument.LoadFromFile carga en memoria los archivos de hasta **MemoryThreshold** bytes (predeterminado 64 MB). Un archivo más grande se abre para lectura compartida y sus objetos se leen bajo demanda a través de una caché de lectura, de modo que abrir un archivo de 3 GB solo lee sus referencias cruzadas y los objetos que utiliza. Establezca MemoryThreshold en 0 para leer todos los archivos bajo demanda.

```
procedure OpenLargeFile(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // files larger than MemoryThreshold (default 64 MB) are read on demand
    // from the file instead of being loaded into memory
    oPDF.MemoryThreshold := 16 * 1024 * 1024;
    oPDF.LoadFromFile(AFileName);
    WriteLn('Pages: ', oPDF.PageCount, ', read on demand: ',
      oPDF.Parser.StreamBacked);
    WriteLn(string(oPDF.ExtractTextW(0)));
  finally
    oPDF.Free;
  end;
end;
```

Comportamiento

- `Parser.StreamBacked` es True cuando el documento se lee bajo demanda, y `Parser.Size` es el tamaño del archivo como Int64.
- El archivo permanece abierto hasta que se carga otro documento o se libera el documento. Mientras tanto, otros procesos pueden leerlo, pero no eliminarlo ni sobrescribirlo.
- SaveIncremental funciona en un documento leído bajo demanda y puede escribir en el mismo archivo: los bytes originales se copian por bloques, el archivo se reemplaza y el analizador vuelve a abrir el nuevo archivo.
- LoadFromStream y LoadFromBytes siempre cargan en memoria. Para leer un stream bajo demanda, utilice el analizador directamente: `TsgcPDFParser.LoadFromStream(AStream, AOwnsStream)`. El stream debe permanecer activo y sin cambios mientras se utiliza el analizador.
- Un proceso de 32 bits no puede contener 2 GB en un solo array: cargar un origen así en memoria genera una excepción que solicita la lectura bajo demanda.

Vea también

- Guardado incremental y salida compacta
- Gestión de errores y entradas hostiles

FDF y XFDF

Los archivos FDF (ISO 32000, 12.7.8) y XFDF (ISO 19444-1, XML) contienen únicamente los valores de los campos de formulario de un PDF. Utilícelos para enviar datos rellenados a un servidor, para rellenar previamente un formulario desde una base de datos o para trasladar valores entre dos versiones de un formulario. TsgcPDFAcroForm (`TsgcPDFDocument.AcroForm`) exporta e importa ambos formatos.

Exportación

```
procedure ExportFormData(const APDFFile, AFDFFile, AXFDFFile: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    // the optional second parameter names the PDF the data belongs to
    oPDF.AcroForm.ExportFDF(AFDFFile, ExtractFileName(APDFFile));
    oPDF.AcroForm.ExportXFDF(AXFDFFile, ExtractFileName(APDFFile));
  finally
    oPDF.Free;
  end;
end;
```

Método	Descripción
ExportFDF(AStream or AFileName; ASourceFile)	Escribe un archivo FDF con el valor de cada campo bajo su nombre completamente cualificado. ASourceFile (opcional) se escribe como /F.
ExportXFDF(AStream or AFileName; ASourceFile)	Escribe un archivo XFDF (UTF-8). ASourceFile (opcional) se escribe como <f href>.

Importación

```
procedure ImportFormData(const APDFFile, AXFDFFile, AOutput: string);
var
  oPDF: TsgcPDFDocument;
  vCount: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.LoadFromFile(APDFFile);
    vCount := oPDF.AcroForm.ImportXFDF(AXFDFFile);
    WriteLn(vCount, ' values set, ', oPDF.AcroForm.ImportUnknownCount,
      ' unknown fields');
    // the appearances are regenerated from the imported values
    oPDF.SaveToFile(AOutput);
  finally
    oPDF.Free;
  end;
end;
```

`ImportFDF` e `ImportXFDF` establecen el valor de cada campo que coincide con su nombre completamente cualificado (por ejemplo `address.city`) y devuelven el número de valores establecidos. Los nombres que no coinciden con ningún campo se cuentan en `ImportUnknownCount`. Las apariencias de los campos modificados se regeneran al guardar el documento.
Los valores de texto son Unicode en ambos formatos, las casillas de verificación y los botones de opción utilizan su valor de exportación (por ejemplo Off o Yes), y las listas de selección múltiple escriben un valor por cada opción seleccionada.

Vea también

- Valores y apariencias de formulario
- Formularios interactivos

Valores y apariencias de formulario

```
procedure CreateForm(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oList: TsgcPDFFormField;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.AddPage(595, 842);
    oPDF.AcroForm.AddTextField('name', 0, 50, 740, 300, 760);
    oPDF.AcroForm.AddCheckBox('agree', 0, 50, 700, 65, 715);
    oList := oPDF.AcroForm.AddListBox('languages', 0, 50, 600, 200, 680,
      ['Delphi', 'C++', 'Pascal', 'Python']);
    // multi-select list box: flag bit 22 (MultiSelect)
    oList.Flags := oList.Flags or $200000;
    oList.SetValues(['Delphi', 'Pascal']);
    // Cyrillic value: the appearance embeds a font that covers it
    oPDF.AcroForm.AppearanceFontName := 'Arial';
    oPDF.AcroForm.FillField('name', 'Иван Петров');
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Listas de selección múltiple

Una lista con el indicador MultiSelect (bit 22 de `Flags`, valor $200000) puede contener varios valores. `SetValues` los selecciona, `ValueCount` y `GetValues(AIndex)` los leen, y `Value` devuelve el primero. Los valores se escriben como un array en /V y como los índices /I de las opciones seleccionadas, y se conservan en FDF y XFDF.

Apariencias Unicode

La apariencia de un valor de campo con caracteres fuera de la codificación de fuente estándar (cirílico, griego, CJK, árabe, hebreo...) se dibuja con una fuente TrueType u OpenType incrustada, registrada como `/SgcU` en los recursos de apariencia y en el diccionario /DR del formulario, de modo que el valor se muestra e imprime correctamente también en visores que no reconstruyen las apariencias.

Propiedad	Descripción
AppearanceFontFile: string	Archivo de fuente (.ttf, .otf) utilizado para estos valores. Tiene prioridad sobre AppearanceFontName.
AppearanceFontName: WideString	Fuente instalada utilizada para estos valores.

Cuando ambos están vacíos, se utiliza la primera fuente instalada entre Arial Unicode MS, Segoe UI, Arial, Yu Gothic, MS Gothic, Microsoft YaHei, Malgun Gothic y Noto Sans que cubra los caracteres. En Linux o en un servidor sin estas fuentes, establezca AppearanceFontFile.

NeedAppearances

`AcroForm.NeedAppearances` escribe `/NeedAppearances true`, lo que pide al visor que reconstruya las apariencias de los campos al abrir el archivo. Es True de forma predeterminada para los formularios nuevos. Los documentos cargados conservan el valor del archivo. sgcPDF siempre escribe sus propias apariencias, de modo que los campos

también se ven correctamente en visores que ignoran el indicador; establézcalo en False para archivos PDF/A, ya que los estándares de PDF/A no permiten que sea true.

Vea también

- FDF y XFDF
- Formularios interactivos

Firmas PAdES

PAdES (ETSI EN 319 142-1) es el perfil europeo de firmas PDF utilizado para las firmas electrónicas avanzadas y cualificadas conforme a eIDAS. TsgcPDFSignatureHandler crea firmas PAdES baseline B-B con CryptoAPI de Windows, acepta un CMS creado por cualquier firmante externo, y agrega sellos de tiempo de documento y datos de validación para los niveles B-T, B-LT y B-LTA.

SubFilter

SubFilter (TsgcPDFSignatureSubFilter)	Se escribe como	Uso
pssPKCS7Detached (predeterminado)	/adbe.pkcs7.detached	Firmas PDF clásicas, compatibles con todos los lectores.
pssCAdESDetached	/ETSI.CAdES.detached	Firmas PAdES baseline. El CMS lleva el atributo ESS signing-certificate-v2.

```
procedure SignPAdES(const AInput, AOutput, APFXFile: string);
var
  oSigner: TsgcPDFSignatureHandler;
begin
  oSigner := TsgcPDFSignatureHandler.Create;
  try
    oSigner.LoadCertificateFromPFX(APFXFile, 'test123');
    // PAdES baseline: /SubFilter /ETSI.CAdES.detached
    oSigner.SubFilter := pssCAdESDetached;
    oSigner.HashAlgorithm := shaSHA256;
    oSigner.SignerName := 'John Smith';
    oSigner.Reason := 'Approved';
    oSigner.Location := 'Barcelona';
    oSigner.SetField('Signature1', 0, 350, 50, 550, 110);
    oSigner.SignDocument(AInput, AOutput);
  finally
    oSigner.Free;
  end;
end;
```

Firmantes externos (OnExternalSign)

Cuando la clave privada no está disponible para CryptoAPI (un servicio de firma remoto, un HSM, un middleware de tarjeta inteligente, una bóveda de claves en la nube), asigne **OnExternalSign**. No es necesario cargar ningún certificado: sgcPDF prepara el diccionario de firma y los rangos de bytes, y el evento devuelve el CMS.

```
procedure TSignEvents.DoExternalSign(Sender: TObject; const AData: TBytes;
  const ADigest: TBytes; var ASignature: TBytes);
begin
  // AData: the bytes covered by /ByteRange, ADigest: their SHA-256 digest.
  // Return a complete DER encoded detached CMS SignedData
  ASignature := SignWithRemoteService(AData);
end;

procedure SignExternal(const AInput, AOutput: string; AEvents: TSignEvents);
var
  oSigner: TsgcPDFSignatureHandler;
begin
  oSigner := TsgcPDFSignatureHandler.Create;
  try
    // no certificate is loaded: the CMS comes from the event
```

```
    oSigner.OnExternalSign := AEvents.DoExternalSign;
    oSigner.SignerName := 'Remote signer';
    oSigner.SetInvisible;
    oSigner.SignDocument(AInput, AOutput);
  finally
    oSigner.Free;
  end;
end;
```

Paráme-tro	Descripción
AData	Los bytes cubiertos por /ByteRange (el archivo sin el marcador de posición /Contents).
ADigest	El resumen (digest) de AData con HashAlgorithm, para servicios que firman un hash.
ASignatu-re	Devuelva un CMS SignedData separado, completo y codificado en DER. Debe caber en Pla-ceholderSize bytes (predeterminado 8192, auméntelo para cadenas de certificados largas o se-llos de tiempo incrustados).

Para firmas PAdES completas construidas con sgcSign (cualquier proveedor de claves, sellos de tiempo de firma, OCSP), utilice el puente TsgcPDFPAdESSigner.

Vea también

- Sellos de tiempo y validación a largo plazo
- Verificación de firmas
- Firmas digitales
- TsgcPDFSignatureHandler

Sellos de tiempo y validación a largo plazo

Una firma solo se puede validar mientras su certificado sea válido y se pueda comprobar su estado de revocación. La validación a largo plazo (LTV) mantiene una firma verificable después de eso, incrustando los datos de validación en el documento y protegiendo todo con un sello de tiempo. TsgcPDFSignatureHandler escribe ambos como actualizaciones incrementales, de modo que las firmas anteriores permanecen válidas.

Sellos de tiempo de documento

TimestampDocument (y TimestampDocumentStream) agrega una firma invisible de tipo `/DocTimeStamp` con SubFilter `/ETSI.RFC3161`. El token de sello de tiempo procede del evento **OnTimestamp**: sgcPDF pasa el resumen (digest) a sellar, y su código envía una solicitud RFC 3161 a una autoridad de sellado de tiempo (TSA) y devuelve el TimeStampToken de la respuesta.

```
procedure TSignEvents.DoTimestamp(Sender: TObject; const ADigest: TBytes;
  AHashAlgorithm: TsgcPDFSignatureHashAlgorithm; var AToken: TBytes);
begin
  // send an RFC 3161 request with ADigest to your time-stamp authority and
  // return the TimeStampToken of the response (DER)
  AToken := RequestTimeStampToken(ADigest, AHashAlgorithm);
end;

procedure AddDocumentTimestamp(const AInput, AOutput: string;
  AEvents: TSignEvents);
var
  oSigner: TsgcPDFSignatureHandler;
begin
  oSigner := TsgcPDFSignatureHandler.Create;
  try
    oSigner.OnTimestamp := AEvents.DoTimestamp;
    // appends a /DocTimeStamp (ETSI.RFC3161) as an incremental update
    oSigner.TimestampDocument(AInput, AOutput);
  finally
    oSigner.Free;
  end;
end;
```

OnTimestamp se utiliza únicamente para los sellos de tiempo de documento. Los sellos de tiempo de firma dentro del CMS (PAdES B-T) los agrega TsgcPDFPAdESSigner.

Document Security Store (DSS)

El DSS (`/DSS` en el catálogo) contiene los certificados, las respuestas OCSP y las CRL que necesita un validador, además de una entrada `/VRI` por cada firma del documento.

```
procedure AddValidationData(const AInput, AOutput: string);
var
  oSigner: TsgcPDFSignatureHandler;
begin
  oSigner := TsgcPDFSignatureHandler.Create;
  try
    // certificates of the chain, CRLs and OCSP responses (DER)
    oSigner.AddDSSCertificate(LoadBytes(Pki + 'ica.der'));
    oSigner.AddDSSCertificate(LoadBytes(Pki + 'root.der'));
    oSigner.AddDSSCRL(LoadBytes(Pki + 'ica.crl'));
    oSigner.AddDSSCRL(LoadBytes(Pki + 'root.crl'));
    // writes /DSS with a /VRI entry per signature (incremental update)
    oSigner.AddDSSDocument(AInput, AOutput);
  finally
    oSigner.Free;
  end;
```

```
end;
```

Método	Descripción
AddDSSCertificate(AData)	Agrega un certificado DER (CA intermedia, CA raíz, certificado de TSA).
AddDSSOCSPResponse(AData)	Agrega una respuesta OCSP DER.
AddDSSCRL(AData)	Agrega una lista de revocación de certificados (CRL) DER.
ClearDSS	Borra los datos recopilados.
AddDSSDocument(AInputFile, AOutputFile), AddDSSStream(AInput, AOutput)	Escribe el DSS como una actualización incremental de un documento firmado.

Niveles

Nivel	Cómo
B-B	Firme con SubFilter pssCAdESDetached.
B-T	B-B con un sello de tiempo de firma, o un sello de tiempo de documento agregado justo después de firmar.
B-LT	Agregue el DSS con la cadena y los datos de revocación.
B-LTA	Agregue un sello de tiempo de documento después del DSS. Repita el DSS y el sello de tiempo antes de que caduque el certificado del sello de tiempo.

La salida de los ejemplos de esta página se validó con pyHanko: la firma CAdES está intacta, es válida y de confianza, las modificaciones posteriores se reconocen únicamente como actualizaciones LTA, y el sello de tiempo de documento está intacto y es de confianza.

Vea también

- Firmas PAdES
- TsgcPDFPAdESSigner
- Verificación de firmas

Verificación de firmas

TsgcPDFSignatureVerifier comprueba todas las firmas y sellos de tiempo de documento de un PDF. `GetSignature(AIndex)` devuelve un TsgcPDFSignatureInfo con los detalles de cada una.

```
procedure VerifyAll(const AFileName: string);
var
  oVerifier: TsgcPDFSignatureVerifier;
  oSig: TsgcPDFSignatureInfo;
  i: Integer;
begin
  oVerifier := TsgcPDFSignatureVerifier.Create;
  try
    oVerifier.LoadDocument(AFileName);
    for i := 0 to oVerifier.GetSignatureCount - 1 do
    begin
      oVerifier.VerifySignature(i);
      oSig := oVerifier.GetSignature(i);
      WriteLn(string(oSig.FieldName), ' ', string(oSig.SubFilter));
      WriteLn('  doc time-stamp: ', oSig.IsDocTimeStamp);
      WriteLn('  signer: ', string(oSig.CertInfo.SubjectName));
      WriteLn('  signed: ', DateTimeToStr(oSig.SignDate));
      WriteLn('  valid: ', oSig.IsValid, ', trusted: ', oSig.IsTrusted,
        ', modified: ', oSig.IsModified);
    end;
  finally
    oVerifier.Free;
  end;
end;
```

Propiedad	Descripción
FieldName	Nombre del campo de firma.
SubFilter	Por ejemplo adbe.pkcs7.detached, ETSI.CAdES.detached o ETSI.RFC3161.
IsDocTimeS-tamp	True para un sello de tiempo de documento (/Type /DocTimeStamp). Su token se verifica con respecto a los bytes cubiertos.
SignDate	La entrada /M de la firma, o el genTime (UTC) del token de un sello de tiempo de documento.
CertInfo	SubjectName, IssuerName, SerialNumber, NotBefore, NotAfter y Thumbprint del certificado del firmante (o de la TSA).
IsValid	La firma CMS coincide con los rangos de bytes firmados.
IsTrusted	El certificado del firmante encadena con una raíz de confianza para Windows. Se comprueba por separado: una firma válida de un firmante desconocido es válida pero no de confianza.
IsModified	La firma no cubre todo el archivo. Las actualizaciones incrementales después de firmar (relleno de formularios, anotaciones, DSS, sellos de tiempo) la establecen.
Name, Reason, Location	Los valores del diccionario de firma.

VerifySignature(AIndex) devuelve IsValid. IsSignatureTrusted(AIndex) e IsDocumentModified(AIndex) devuelven los otros dos resultados. La confianza se evalúa con el almacén de certificados de Windows, por lo que los certifi-

cados de prueba se reportan como no confiables. Para la validación de políticas PAdES con respecto a una lista de confianza específica, utilice un validador dedicado como sgcSign.

Vea también

- Firmas PAdES
- TsgcPDFSignatureVerifier

TsgcPDFPAdESSigner (puente sgcSign)

TsgcPDFPAdESSigner (unidad **sgcPDF_Sign_sgcSign**) crea firmas PAdES baseline B-B, B-T, B-LT y B-LTA con **sgcSign**, la biblioteca de firma digital de eSeGeCe. sgcPDF gestiona el lado del PDF (campo, apariencia, actualizaciones incrementales, DSS, sello de tiempo de documento) y sgcSign gestiona la criptografía: el CMS con los atributos que exige PAdES, el proveedor de claves, y los clientes de TSA y OCSP.
Dado que se accede a la clave privada mediante un proveedor de claves de sgcSign, funciona cualquier almacén de claves que sgcSign admita: archivos PFX y PEM, tokens PKCS#11 y tarjetas inteligentes, el almacén de certificados de Windows y HSM en la nube.

Requisitos

- Una licencia de sgcSign y su carpeta Source en la ruta de búsqueda del proyecto, junto a sgcPDF.
- sgcPDF_Sign_sgcSign.pas es una unidad opcional: **no** forma parte de los paquetes de sgcPDF. Agréguela a su proyecto.

Ejemplo

```
procedure SignPAdESLT(const AInput, AOutput: string);
var
  oKey: TsgcPEMKeyProvider;
  oSigner: TsgcPDFPAdESSigner;
begin
  oKey := TsgcPEMKeyProvider.Create(nil);
  oSigner := TsgcPDFPAdESSigner.Create(nil);
  try
    // any sgcSign key provider: PEM, PFX, PKCS#11, Windows store, cloud HSM
    oKey.LoadFromFile(Pki + 'signer.pem', Pki + 'signer.key');
    oSigner.KeyProvider := oKey;
    oSigner.Level := paBLT;
    oSigner.TSAURL := 'http://timestamp.digicert.com';
    // validation material: CA certificates and CRLs (or an OCSPClient)
    oSigner.AddCertificate(LoadDER(Pki + 'ica.der'));
    oSigner.AddCertificate(LoadDER(Pki + 'root.der'));
    oSigner.AddCRL(LoadDER(Pki + 'ica.crl'));
    oSigner.AddCRL(LoadDER(Pki + 'root.crl'));
    // signature field and appearance through the sgcPDF handler
    oSigner.Handler.SetField('Signature1', 0, 350, 50, 550, 110);
    oSigner.Handler.SignerName := 'John Smith';
    oSigner.Handler.Reason := 'Approved';
    oSigner.SignPDFFile(AInput, AOutput);
    oSigner.KeyProvider := nil;
  finally
    oSigner.Free;
    oKey.Free;
  end;
end;
```

Este ejemplo se compiló con Delphi 13 contra las fuentes de sgcPDF y sgcSign y se ejecutó con una TSA pública real: pyHanko valida el resultado como una firma ETSI.CAdES.detached intacta y de confianza, con un sello de tiempo de firma válido y un /DSS.

Miembro	Descripción
KeyProvider: IsgcKeyProvider	El proveedor de claves de sgcSign con el certificado del firmante y la clave privada.
Level: TsgcPDFPAdESLevel	paBB, paBT, paBLT o paBLTA. Predeterminado paBB.

Miembro	Descripción
TSAURL: string / TSAClient: Isgc-TSAClient	Autoridad de sellado de tiempo para B-T y superiores: una URL (se utiliza un cliente TSA interno de sgcSign) o su propio cliente.
OCSPClient: IsgcOCSPClient	Opcional. Cuando se asigna, la respuesta OCSP del certificado del firmante se obtiene y se almacena para B-LT.
HashAlgorithm	shaSHA256 (predeterminado), shaSHA384 o shaSHA512.
AddCertificate, AddCRL, AddOCS-PResponse, ClearValidationData	Material de validación. Los certificados también se incrustan en el CMS; las CRL y las respuestas OCSP van al /DSS.
Handler: TsgcPDFSignatureHandler	Campo de firma, apariencia, motivo, lugar y tamaño del marcador de posición.
SignPDF(AInput, AOutput), SignPDFFile(AInputFile, AOutputFile)	Firma en el nivel seleccionado.
ExtendToLTA(AInput, AOutput)	Agrega el DSS y un sello de tiempo de documento a un PDF ya fir-mado.

El puente no escribe un atributo signing-time de CMS, que PAdES prohíbe junto a la entrada /M del diccionario de firma.

Vea también

- Firmas PAdES
- Sellos de tiempo y validación a largo plazo

Cifrado con certificado

El cifrado con contraseña comparte un único secreto con todos los lectores. El cifrado con certificado (clave pública), el manejador de seguridad `/Adobe.PubSec`, en cambio cifra la clave del documento para el certificado X.509 de cada destinatario: solo los propietarios de las claves privadas correspondientes pueden abrir el archivo, no se intercambia ninguna contraseña, y cada destinatario puede obtener permisos diferentes.

Cifrado

```
procedure EncryptForRecipients(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.AddPage(595, 842).DrawText('For the finance team only', 50, 780,
      sfHelvetica, 14);
    // every recipient gets its own permissions
    oPDF.AddEncryptionRecipient(Pki + 'signer.der', [ppPrint, ppCopy,
      ppExtract]);
    oPDF.AddEncryptionRecipient(Pki + 'tsa.pem', [ppPrint]);
    // emAES_256 writes /Adobe.PubSec with /SubFilter /adbe.pkcs7.s5
    oPDF.SetCertificateEncryption(emAES_256);
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Método	Descripción
AddEncryptionRecipient(ACertificate: TBytes; APermissions)	Agrega un destinatario a partir de un certificado DER o PEM.
AddEncryptionRecipient(ACertificateFile: string; APermissions)	Igual, desde un archivo .cer, .der o .pem.
AddEncryptionRecipientFromStore(ACertificateId; APermissions; AStoreName)	Un certificado del almacén de Windows, por huella digital SHA-1 (hexadecimal) o parte del asunto. AStoreName es MY de forma predeterminada.
SetCertificateEncryption(AMethod)	Activa el cifrado para los destinatarios agregados. emAES_256 y emAES_128 escriben /SubFilter /adbe.pkcs7.s5 (filtros de cifrado), emRC4_128 y emRC4_40 el /adbe.pkcs7.s4 heredado.

APermissions es un conjunto de TsgcPDFPermission: ppPrint, ppModify, ppCopy, ppAnnotate, ppFillForms, ppExtract, ppAssemble y ppPrintHigh.

Apertura

```
procedure OpenWithCertificate(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    // the private key of one recipient opens the document
```

```
    oPDF.SetDecryptionCertificate(Pki + 'signer.pfx', 'test123');
    oPDF.LoadFromFile(AFileName);
    WriteLn(string(oPDF.ExtractTextW(0)));
    WriteLn('Can print: ', ppPrint in oPDF.Encryption.Permissions);
    WriteLn('Can modify: ', ppModify in oPDF.Encryption.Permissions);
  finally
    oPDF.Free;
  end;
end;
```

Establezca la clave privada antes de LoadFromFile o LoadFromStream con `SetDecryptionCertificate(APFXFileName, APassword)`, `SetDecryptionCertificate(APFX: TBytes, APassword)` o `SetDecryptionCertificateFromStore(AStoreName)`, que prueba los certificados con clave privada del almacén del usuario actual. `Encryption.Permissions` contiene entonces los permisos del destinatario que abrió el archivo.

Plataforma

El cifrado y descifrado con certificado utilizan CryptoAPI de Windows (crypt32.dll, parte de Windows), por lo que solo están disponibles en Windows. El archivo producido en el ejemplo fue descifrado de forma independiente por pyHanko con el PFX del primer destinatario.

Notas de seguridad

- Las claves de archivo, los vectores de inicialización AES y las sales provienen del CSPRNG del sistema operativo (CryptGenRandom en Windows, /dev/urandom en Linux, macOS, iOS y Android).
- emAES_256 escribe la revisión 6 del manejador de seguridad de PDF 2.0 (ISO 32000-2). Los archivos con la revisión 5 obsoleta (Acrobat 9) se siguen abriendo.
- El cifrado solo se puede cambiar mediante un guardado completo: SaveIncremental genera una excepción cuando se cambió.

Vea también

- Cifrado
- TsgcPDFRedactor

TsgcPDFRedactor

TsgcPDFRedactor (unidad sgcPDF_Redaction) realiza una redacción real: el texto, los píxeles de imagen, los gráficos vectoriales y las anotaciones dentro de las áreas marcadas se eliminan del archivo, no solo se cubren con un recuadro negro. El contenido redactado no se puede seleccionar, copiar, extraer ni recuperar de la salida.

```
procedure RedactFile(const AInput, AOutput: string);
var
  oRedactor: TsgcPDFRedactor;
  vOptions: TsgcPDFRedactionOptions;
begin
  oRedactor := TsgcPDFRedactor.Create;
  try
    // every occurrence of a text, on all pages
    oRedactor.AddText('John Smith');
    // an area of page 0: left, bottom, right, top in points
    oRedactor.AddArea(0, 45, 735, 300, 752);
    oRedactor.AddArea(0, 50, 500, 250, 650);
    vOptions := oRedactor.Options;
    vOptions.OverlayText := 'REDACTED';
    vOptions.FillColorR := 0;
    vOptions.FillColorG := 0;
    vOptions.FillColorB := 0;
    vOptions.RemoveMetadata := True;
    oRedactor.Options := vOptions;
    oRedactor.Apply(AInput, AOutput);
    WriteLn('Text matches: ', oRedactor.TextMatches);
    WriteLn('Removed glyphs: ', oRedactor.RemovedGlyphs);
    WriteLn('Removed paths: ', oRedactor.RemovedPaths);
  finally
    oRedactor.Free;
  end;
end;
```

Marcado

Método	Descripción
AddArea(APageIndex; ARect: TsgcPDFRect)	Un rectángulo en puntos, en el espacio de usuario predeterminado de la página (origen en la parte inferior izquierda, antes del /Rotate de la página).
AddArea(APageIndex; ALeft, ABottom, ARight, ATop)	Igual, con coordenadas.
AddQuad(APageIndex; AQuad: TsgcPDFRedactionQuad)	Cualquier cuadrilátero convexo, para texto rotado. sgcPDFRedaction-QuadFromRect convierte un rectángulo.
AddText(AText; ACaseSensitive; AWholeWord; APageIndex)	Cada aparición de AText en una página, o en todas las páginas cuando APageIndex es -1. Las coincidencias se buscan dentro de una línea de texto, y las secuencias de espacio en blanco coinciden con cualquier espacio en blanco.
Clear	Elimina todas las marcas.

Aplicación

Método	Descripción
Apply(ADocument: TsgcPDFDocument)	Redacta un documento cargado: se guarda, se redacta y se vuelve a cargar a partir de los bytes redactados.
Apply(AInputFile, AOutputFile)	Redacta un archivo.
ApplyToStream(AInput, AOutput), ApplyToBytes(AData)	Redacta un stream o un array de bytes.

La salida siempre es un archivo nuevo completo (un guardado completo): los bytes originales, que una actualización incremental conservaría, nunca se escriben. Después de Apply están disponibles las estadísticas TextMatches, RemovedGlyphs, RedactedImages, RemovedImages, RemovedPaths y RemovedAnnotations, y `GetLog` devuelve un registro de la operación.

Opciones

Campo de TsgcPDFRedactionOptions	Descripción
FillColorR, FillColorG, FillColorB	Color (0 a 1) de los rectángulos pintados sobre las áreas. Predeterminado negro.
OverlayText, OverlayFontSize	Texto dibujado centrado en cada área (Helvetica), 0 ajusta al área.
RemoveAnnotationsInArea	Elimina las anotaciones cuyo rectángulo interseca un área, incluidos los enlaces y los widgets de campo de formulario con sus campos. Predeterminado True.
RemoveMetadata	Elimina el diccionario /Info y los metadatos XMP.
RemoveAttachments	Elimina los archivos incrustados y las anotaciones de archivo adjunto.
RemoveJavaScript	Elimina JavaScript, /OpenAction, las acciones adicionales y las acciones JavaScript, Launch, SubmitForm e ImportData.

Documentos cifrados

Abra el documento con su contraseña y llame a `Apply(ADocument)`. El contenido se redacta descifrado y el documento conserva su configuración de cifrado, de modo que el siguiente SaveToFile o SaveToStream lo escribe cifrado de nuevo. SaveIncremental genera una excepción después de una redacción. Las sobrecargas de archivo, stream y bytes rechazan la entrada cifrada, ya que no aceptan contraseña.

```
procedure RedactEncrypted(const AFileName, APassword, AOutput: string);
var
  oPDF: TsgcPDFDocument;
  oRedactor: TsgcPDFRedactor;
begin
  oPDF := TsgcPDFDocument.Create;
  oRedactor := TsgcPDFRedactor.Create;
  try
    oPDF.SetPassword(AnsiString(APassword));
    oPDF.LoadFromFile(AFileName);
    oRedactor.AddText('Salary', False, True);
    oRedactor.Apply(oPDF);
    // full save: written encrypted again with the same settings
```

```
    oPDF.SaveToFile(AOutput);
  finally
    oRedactor.Free;
    oPDF.Free;
  end;
end;
```

Cómo se elimina el contenido

- **Texto**: cada glifo cuyo cuadro interseca un área se elimina del stream de contenido; los glifos restantes conservan sus posiciones.
- **Imágenes**: las imágenes que sgcPDF puede decodificar, con 1 a 8 bits por componente, se reescriben con los píxeles redactados pintados en el color de relleno (RedactedImages). Las imágenes que no se pueden decodificar, por ejemplo con un filtro no admitido, demasiado grandes o con 16 bits por componente, se eliminan por completo (RemovedImages).
- **Gráficos vectoriales**: un subtrayecto que se encuentra completamente dentro de un área se elimina (RemovedPaths). Los subtrayectos que solo cruzan un área permanecen, bajo el rectángulo de relleno. Los trayectos de recorte no se modifican.
- Los **XObjects de formulario** se redactan de forma recursiva. Un XObject de formulario anidado demasiado profundamente o con un filtro no admitido se elimina por completo.
- Una página cuyo stream de contenido utiliza un filtro no admitido genera EsgcPDFRedactionError en lugar de quedar sin redactar.

En el ejemplo, pypdf no encuentra rastro de las palabras redactadas en la salida, ni en el texto extraído ni en ningún stream de contenido, y el archivo tiene una sola revisión.

Vea también

- Cifrado con certificado
- Extracción de texto

PDF etiquetado

Un PDF etiquetado (ISO 32000-1, 14.8) contiene un árbol de estructura lógica junto al contenido de la página: encabezados, párrafos, listas, tablas, figuras con texto alternativo y enlaces, en orden de lectura. Los lectores de pantalla, el reflujo en dispositivos móviles y la exportación de texto lo utilizan, y es un requisito de la conformidad con PDF/UA y con el nivel A de PDF/A.

Construcción de la estructura

Establezca **TsgcPDFDocument.Tagged** en True y envuelva las llamadas de dibujo de cada página entre `Page.Tagging.BeginElement` y `EndElement`. El etiquetado de página escribe los operadores de contenido marcado (BDC/EMC con MCID) y los vincula al elemento; `StructureTree` recopila los elementos, el ParentTree y el RoleMap, y escribe /MarkInfo, /Lang y los /StructParents de la página al guardar el documento.

```
procedure CreateTaggedPDF(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oPage: TsgcPDFPage;
  oTag: TsgcPDFPageTagging;
  oFigure, oCell: TsgcPDFStructElement;
  vFont: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.Info.Title := 'Quarterly report';
    oPDF.Tagged := True;
    oPDF.StructureTree.Lang := 'en-US';
    vFont := oPDF.AddTrueTypeFont('Arial', False, False);
    oPage := oPDF.AddPage(595, 842);
    oTag := oPage.Tagging;
    // running header: not part of the logical content
    oTag.BeginArtifact(akPagination, asHeader);
    oPage.DrawTextTT('ACME Corp.', 50, 810, vFont, 8);
    oTag.EndArtifact;
    oTag.BeginElement(stH1);
    oPage.DrawTextTT('Quarterly report', 50, 770, vFont, 20);
    oTag.EndElement;
    oTag.BeginElement(stP);
    oPage.DrawTextTT('Sales grew in every region.', 50, 740, vFont, 11);
    oTag.EndElement;
    // figures need an alternate text
    oFigure := oTag.BeginElement(stFigure);
    oFigure.Alt := 'Blue bar chart of the sales per region';
    oFigure.SetBBox(50, 600, 250, 700);
    oPage.DrawRectangleFill(50, 600, 200, 100, 0.2, 0.4, 0.8);
    oTag.EndElement;
    // grouping elements (no content) are opened on the tree
    oPDF.StructureTree.BeginElement(stTable);
    oPDF.StructureTree.BeginElement(stTR);
    oCell := oTag.BeginElement(stTH);
    oCell.Scope := tsColumn;
    oPage.DrawTextTT('Region', 50, 570, vFont, 11);
    oTag.EndElement;
    oCell := oTag.BeginElement(stTH);
    oCell.Scope := tsColumn;
    oPage.DrawTextTT('Sales', 200, 570, vFont, 11);
    oTag.EndElement;
    oPDF.StructureTree.EndElement;
    oPDF.StructureTree.BeginElement(stTR);
    oTag.BeginElement(stTD);
    oPage.DrawTextTT('North', 50, 550, vFont, 11);
    oTag.EndElement;
```

```
    oTag.BeginElement(stTD);
    oPage.DrawTextTT('1200', 200, 550, vFont, 11);
    oTag.EndElement;
    oPDF.StructureTree.EndElement;
    oPDF.StructureTree.EndElement;
    // a link: the annotation added inside the Link element is attached to it
    oTag.BeginElement(stLink).Alt := 'eSeGeCe web site';
    oPage.DrawTextTT('www.esegece.com', 50, 520, vFont, 11);
    oPDF.Annotations.AddLinkURI(0, 50, 515, 150, 532,
      'https://www.esegece.com');
    oTag.EndElement;
    // PDF/UA-1 and PDF/A-2a in the same file
    oPDF.SetPDFALevel(paA2a);
    oPDF.PDFUA := True;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Miembro	Descripción
Page.Tagging.BeginElement(AType; ATitle)	Abre un elemento cuyo contenido se dibuja en esta página, como hijo del elemento actual. Devuelve el TsgcPDFStructElement.
StructureTree.BeginElement(AType; ATitle)	Abre un elemento de agrupación sin contenido de página (Table, TR, L, Sect, Part...).
EndElement	Cierra el elemento actual (en el etiquetado de página o en el árbol, según corresponda a la llamada Begin).
BeginCustomElement(ATypeName...) and StructureTree.AddRole	Tipos de estructura personalizados, asignados a un tipo estándar mediante /RoleMap.
Page.Tagging.BeginArtifact(AKind; ASubtype), EndArtifact	Contenido que no forma parte del documento: encabezados y pies de página repetidos, números de página, marcas de agua, decoración. AKind: akPagination, akLayout, akPage, akBackground. ASubtype: asHeader, asFooter, asWatermark.
StructureTree.Lang	Idioma natural del documento (catálogo /Lang), por ejemplo en-US.

TsgcPDFStructType tiene todos los tipos estándar: stDocument, stPart, stArt, stSect, stDiv, stH, stH1 a stH6, stP, stL, stLI, stLbl, stLBody, stTable, stTHead, stTBody, stTFoot, stTR, stTH, stTD, stFigure, stCaption, stLink, stSpan, stQuote, stNote, stReference, stCode, stBlockQuote, stFormula, stTOC, stTOCI y stAnnot.

Propiedades del elemento

Propiedad	Descripción
Alt	Texto alternativo. Obligatorio para Figure y Formula: guardar un Figure sin Alt genera EsgcPDFTaggedError.
ActualText, Expansion, Lang, Title, ID	Texto de reemplazo, expansión de abreviatura, idioma del elemento, título e identificador.
SetBBox(ALeft, ABottom, ARight, ATop)	Cuadro delimitador de un Figure, Formula o Table.

Propiedad	Descripción
Scope, RowSpan, ColSpan, Headers	Celdas de tabla: alcance de un TH (tsRow, tsColumn, tsBoth), extensiones e identificadores de las celdas de encabezado.

Enlaces y anotaciones

Una anotación agregada con `Annotations.AddLinkURI` (u otro método Add) mientras un elemento Link está abierto se adjunta a él como una referencia de objeto, con su /StructParent. `Page.Tagging.AddAnnotation` adjunta una explícitamente.

Tablas

Un TsgcPDFTable dibujado en un documento etiquetado se etiqueta automáticamente como Table, THead, TBody, TR, TH y TD. Con DrawMultiPage, las filas de encabezado repetidas en las páginas siguientes se etiquetan como artefactos de paginación, de modo que un lector de pantalla las lee una sola vez.

```
procedure CreateTaggedTable(const AFileName: string);
var
  oPDF: TsgcPDFDocument;
  oTable: TsgcPDFTable;
  i: Integer;
begin
  oPDF := TsgcPDFDocument.Create;
  try
    oPDF.Tagged := True;
    oPDF.StructureTree.Lang := 'en-US';
    oTable := TsgcPDFTable.Create(2);
    try
      oTable.SetHeaders(['Product', 'Quantity']);
      for i := 1 to 60 do
        oTable.AddRowData([AnsiString('Item ' + IntToStr(i)),
          AnsiString(IntToStr(i))]);
      // tagged as Table / TR / TH / TD; the header repeated on the next
      // pages is tagged as a pagination artifact
      oTable.DrawMultiPage(oPDF, 50, 790, 495, 595, 842);
    finally
      oTable.Free;
    end;
    oPDF.SaveToFile(AFileName);
  finally
    oPDF.Free;
  end;
end;
```

Los documentos etiquetados cargados conservan su árbol de estructura al guardarlos.

Vea también

- PDF/UA y PDF/A nivel A
- Validador de accesibilidad
- Tablas

PDF/UA y PDF/A nivel A

PDF/UA-1

PDF/UA-1 (ISO 14289-1) es el estándar para PDF universalmente accesible, exigido por las leyes de accesibilidad para documentos públicos en muchos países. Establezca **TsgcPDFDocument.PDFUA** en True: el documento se guarda con pdfuaid:part 1 y dc:title en los metadatos XMP y /DisplayDocTitle en las preferencias del visor.
El guardado genera una excepción cuando no se puede cumplir un requisito de PDF/UA:

- el documento no está etiquetado (`Tagged` es False);
- `StructureTree.Lang` está vacío;
- `Info.Title` está vacío;
- el texto se dibuja con una de las 14 fuentes estándar, que no están incrustadas: utilice AddTrueTypeFont (o AddTrueTypeFontFromFile) y DrawTextTT;
- el documento está cifrado sin el permiso ppExtract, que necesita la tecnología de asistencia.

PDF/A nivel A

El nivel A (accesible) de PDF/A añade los requisitos de PDF etiquetado al nivel B. Pase **paA1a**, **paA2a** o **paA3a** a SetPDFALevel, y etiquete el contenido igual que para PDF/UA. PDF/UA y PDF/A se pueden combinar en el mismo archivo:

```
oPDF.SetPDFALevel(paA2a);
oPDF.PDFUA := True;
```

El archivo creado en PDF etiquetado utiliza exactamente esta combinación y pasa veraPDF tanto para PDF/UA-1 como para PDF/A-2a.

TsgcPDFALevel	Estándar
paA1b, paA1a	PDF/A-1 (ISO 19005-1), nivel B o A
paA2b, paA2a	PDF/A-2 (ISO 19005-2), nivel B o A
paA3b, paA3a	PDF/A-3 (ISO 19005-3), nivel B o A. Permite cualquier archivo incrustado.

Lista de comprobación

- Establezca Info.Title y StructureTree.Lang.
- Incruste todas las fuentes (TrueType u OpenType).
- Etiquete todo el contenido: cada texto y gráfico pertenece a un elemento o a un artefacto.
- Dé a cada Figure y Formula un texto Alt.
- Utilice encabezados en orden (H1, luego H2...), celdas TH con Scope en las tablas, y elementos Link para los enlaces.
- Valide el archivo guardado con el validador.

Vea también

- PDF etiquetado
- Validador de accesibilidad
- PDF/A

Validador de accesibilidad

TsgcPDFAValidator (unidad sgcPDF_PDFA) comprueba un archivo guardado con respecto a un nivel de PDF/A. Para los niveles A también comprueba la estructura, y con **PDFUA** establecido en True ejecuta también las comprobaciones de PDF/UA-1.

```
procedure ValidateAccessible(const AFileName: string);
var
  oParser: TsgcPDFParser;
  oValidator: TsgcPDFAValidator;
  i: Integer;
begin
  oParser := TsgcPDFParser.Create;
  oValidator := TsgcPDFAValidator.Create(paA2a);
  try
    oParser.LoadFromFile(AFileName);
    oParser.Parse;
    // PDF/UA-1 checks in addition to the PDF/A-2a ones
    oValidator.PDFUA := True;
    if oValidator.Validate(oParser) then
      WriteLn('PDF/A-2a and PDF/UA-1: no errors')
    else
      for i := 0 to oValidator.Errors.Count - 1 do
        WriteLn('Error: ', oValidator.Errors[i]);
    for i := 0 to oValidator.Warnings.Count - 1 do
      WriteLn('Warning: ', oValidator.Warnings[i]);
  finally
    oValidator.Free;
    oParser.Free;
  end;
end;
```

Qué se comprueba

Método	Comprobaciones
CheckStructure	Árbol de estructura, /MarkInfo /Marked, /Lang.
CheckContentTagged	Todos los operadores de texto y de dibujo de cada página pertenecen a un elemento de estructura o a un artefacto.
CheckStructureElements	Texto alternativo de Figure y Formula, secuencia de encabezados, estructura de tablas y celdas de encabezado.
CheckFontUnicode	Cada fuente tiene un mapa ToUnicode o una codificación estándar, para que el texto se pueda leer.
CheckPDFUA	pdfuaid:part, dc:title, /DisplayDocTitle, las anotaciones en la estructura, /Tabs /S de la página.

Validate ejecuta todas las comprobaciones del nivel (y de PDF/UA cuando está activado) y devuelve True cuando no hay errores. Los métodos individuales se pueden llamar por separado.

Errores y advertencias

`Errors` enumera los problemas que rompen la conformidad. Cada mensaje comienza con la cláusula de la norma a la que se refiere, para que pueda consultarla, por ejemplo:

```
[PDF/A-2a 6.7.3.3] [PDF/UA-1 7.1] Page 1: 1 content operator(s) neither tagged nor marked as artifact (first: TJ)
[PDF/UA-1 5] XMP metadata without pdfuaid:part 1
```

`Warnings` enumera problemas que no son errores de la norma pero que perjudican la accesibilidad, por ejemplo:

```
[PDF/UA-1 7.4.2] Heading level skipped: H1 followed by H3
```

El validador es una comprobación integrada y rápida para su propia salida. No sustituye a un verificador de conformidad completo como veraPDF o PAC para documentos de otro origen.

Vea también

- PDF/UA y PDF/A nivel A
- PDF etiquetado
- PDF/A

Preguntas frecuentes

¿Necesito distribuir una DLL?

No. sgcPDF se compila dentro de su ejecutable. Solo utiliza bibliotecas que forman parte de Windows: GDI para el renderizado y la carga de fuentes, y CryptoAPI para las firmas digitales.

¿Qué plataformas son compatibles?

Windows de 32 y 64 bits con Delphi 7 a Delphi 13 y C++Builder 2007 a 13. El núcleo, el renderizador ráster y el visor FireMonkey también se ejecutan en Linux 64-bit, macOS, iOS y Android. Vea Instalación.

¿Dónde está el origen de las coordenadas de la página?

En la esquina inferior izquierda de la página, en puntos (1/72 de pulgada). El eje Y crece hacia arriba. Vea Arquitectura.

¿Por qué no aparecen los caracteres acentuados o no latinos?

Las fuentes estándar solo cubren el conjunto de caracteres WinAnsi. Utilice una fuente TrueType con AddTrueTypeFont y DrawTextTT. Vea Texto y fuentes.

¿Cómo escribo texto en chino, japonés o coreano?

Agregue una fuente TrueType CJK instalada, por ejemplo el nombre devuelto por sgcPDFCJKFontNameFromEnum(cjkMSGothic), y dibuje con DrawTextTT.

¿Cómo abro un PDF protegido con contraseña?

Llame a SetPassword antes de LoadFromFile. Vea Cifrado.

Mi código de barras no aparece.

Pase el generador de contenido de la página (`Page.Content`) a TsgcPDFBarcodeGenerator, no la página. Vea Códigos de barras.

sgcPDFExtractAllText devuelve una cadena vacía.

Un TsgcPDFParser debe analizarse después de cargarlo: llame a Parse después de LoadFromFile. TsgcPDFDocument.LoadFromFile hace esto automáticamente.

¿Por qué la firma es válida pero no de confianza?

El certificado del firmante no encadena con una raíz de confianza de Windows, lo cual es normal en certificados de prueba autofirmados. Utilice un certificado emitido por una autoridad de certificación de confianza para producción.

El optimizador no redujo mi archivo.

Los archivos que ya están comprimidos y no tienen objetos duplicados no pueden reducirse más. Active DownsampleImages o las opciones Remove para reducirlos aún más. Vea Optimización de archivos PDF.

¿Puedo limitar la memoria utilizada al abrir archivos subidos?

Sí. Reduzca sgcPDFMaxDecodedSize. Vea Gestión de errores y entradas hostiles.

¿sgcPDF se ejecuta en Linux?

Sí. Las unidades del núcleo compilan con el compilador de Delphi para Linux de 64 bits: cree, edite, cifre con contraseñas, extraiga texto y renderice páginas a PNG con TsgcPDFRendererRaster en una aplicación de servidor o de consola. Cargue fuentes desde archivos con AddTrueTypeFontFromFile. Vea Linux.

¿Puedo utilizar sgcPDF desde C++Builder?

Sí. Se incluyen paquetes para C++Builder 2007 a 13 y los encabezados generados están listos para usar. Vea C++Builder.

¿Cómo creo un archivo PDF/A-3 con una factura incrustada (Factur-X, ZUGFeRD)?

Cree el XML de la factura en su aplicación y páselo a Attachments.AddFacturX. El documento se guarda como PDF/A-3b con el XML y los metadatos de Factur-X. Vea Facturas Factur-X y ZUGFeRD.

¿sgcPDF puede crear firmas PAdES?

Sí. Establezca SubFilter en pssCAdESDetached para PAdES B-B, luego agregue datos de validación con AddDSSDocument y un sello de tiempo del documento con TimestampDocument para B-LT y B-LTA. Con sgcSign, TsgcPDFPAdESSigner crea todos los niveles en una sola llamada con cualquier proveedor de claves de sgcSign. Vea Firmas PAdES.

¿Un rectángulo negro oculta el texto que hay debajo?

No, el texto sigue estando en el archivo. Utilice TsgcPDFRedactor, que elimina el texto, los píxeles de imagen y los gráficos bajo las áreas redactadas. Vea TsgcPDFRedactor.

¿Cómo hago un PDF accesible?

Etiquete el contenido con Page.Tagging, establezca Info.Title y StructureTree.Lang, incruste las fuentes y establezca PDFUA en True. Vea PDF etiquetado.

Introducción

sgcReports es un motor de informes sin interfaz gráfica y neutral respecto a la plataforma para Delphi, construido sobre sgcPDF. Está orientado en primer lugar a la generación de informes en servidor y web: servicios de Windows, daemons Linux64 y endpoints HTTP, y se ejecuta sin cambios en aplicaciones de escritorio y móviles VCL y FMX.
El motor produce salida PDF accesible (PDF/UA-1) y PDF de archivo (PDF/A-3) de forma predeterminada. Un informe también se puede exportar a XLSX, CSV o PNG a partir de las mismas páginas preparadas.

Sin interfaz por diseño

El núcleo (modelo de informe, expresiones, fuentes de datos, tipos de letra, diseño, motor y exportaciones) no tiene dependencia de VCL ni de FMX, ni de ninguna DLL externa. Compila y se ejecuta en Windows, Linux64, macOS, iOS y Android. Los componentes de vista previa (`TsgcReportPreviewVCL`, `TsgcReportPreviewFMX`) y el visor de navegador son paquetes complementarios opcionales.

Accesible y de archivo por defecto

La exportación a PDF etiqueta automáticamente el árbol de estructura: encabezados, párrafos, tablas con filas de encabezado, figuras con texto alternativo y artefactos, de modo que la propiedad `PDFUA` es `True` de forma predeterminada. La salida de archivo PDF/A-3 se controla con la propiedad `PDFA` y se puede combinar con PDF/UA. Véase Accesibilidad y PDF/A.

Plataformas

Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13. Windows, Linux64, macOS, iOS y Android a través de FireMonkey. El núcleo compila en Delphi 7 sin genéricos, métodos anónimos, variables en línea ni métodos de registro.

Ediciones y requisitos

Paquete	Requiere	Necesario para
`sgcReports` (núcleo)	sgcPDF	Modelo de informe, expresiones, fuentes de datos, tipos de letra, diseño, motor, exportación PDF/XLSX/CSV/PNG, API fluida, definiciones JSON. sgcPDF es obligatorio en todas las configuraciones.
`sgcReportsVCL` / `sgcReportsFMX`	núcleo, VCL o FMX	`TsgcReportPreviewVCL` o `TsgcReportPreviewFMX`.
`sgcReportsHTML`	núcleo, sgcWebSockets (edición Professional o superior), sgcHTML	`TsgcReportHTTPEndpoint`, `TsgcReportHTMLViewer` y `TsgcReportHTMLDesigner`. sgcWebSockets solo es necesario para este paquete; el núcleo nunca lo usa.

Siguiente: Instalación, y luego Primeros pasos.

Instalación

sgcReports forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia. sgcReports requiere sgcPDF y el instalador se ocupa de ello.

Requisitos

- Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 o 13 (o la versión de C++Builder correspondiente, C2007 a C13).
- sgcPDF, la misma versión de Delphi. El instalador de sgcUI lo instala junto con sgcReports.
- sgcWebSockets edición Professional o superior, y sgcHTML, solo cuando se instala el paquete `sgcReportsHTML`.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcReports (installs sgcPDF)** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcReports

- **sgcPDF**, el motor sobre el que se construye sgcReports. Al seleccionar sgcReports también se selecciona sgcPDF, y sgcPDF se instala siempre que se instala sgcReports.
- **sgcReports HTML para sgcWebSockets**, que aparece en **sgcWebSockets bridges**. Instala los paquetes `sgcReportsHTML` (endpoint HTTP, visor de navegador y diseñador web) para Delphi 10 Seattle y posteriores. El instalador solo lo ofrece cuando sgcWebSockets ya está instalado en el equipo y su licencia incluye sgcReports. De lo contrario aparece atenuado.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado, incluidos sgcReports y sgcPDF. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Reports`	Las demos de sgcReports.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcPDF se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluidos sgcReports y sgcPDF. Una compilación de prueba muestra un aviso de inicio (`sgcReports_License.pas`). Una compilación con licencia no lo hace.
- **Community**: sgcPDF no tiene edición Community, por lo que el instalador gratuito Community de sgcUI no lo instala. Ofrece sgcReports solo cuando ya hay un sgcPDF con licencia instalado en el equipo, ya sea de una instalación registrada de sgcUI que incluya sgcPDF o de una instalación independiente de sgcPDF, y compila sgcReports contra ese sgcPDF. Sin uno, sgcReports aparece atenuado y el instalador le pide que instale primero sgcPDF. Los puentes de sgcWebSockets no forman parte del instalador Community.

Migración desde instalaciones independientes de sgcPDF y sgcReports

Si sgcPDF o sgcReports se instalaron antes con su propio instalador independiente, el instalador de sgcUI detecta esa instalación y muestra sus rutas de bibliotecas y sus paquetes de diseño en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta. La página solo se muestra cuando se encuentra una instalación antigua. Tras la migración, ambos productos se compilan desde la única carpeta `sgcUI\Source`.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcReportsD13.dpk`.

Paquete	Contenido
`sgcPDF<V>` , `sgcPDFVCL<V>` , `sgcPDFFMX<V>` / `dclsgcPDF<V>` , `dclsgcPDFFMX<V>`	sgcPDF, el motor que requiere sgcReports (runtime / diseño).
`sgcReports<V>`	Motor principal y exportaciones (runtime).
`sgcReportsVCL<V>` / `dclsgcReports<V>`	Vista previa VCL (runtime / diseño).
`sgcReportsFMX<V>` / `dclsgcReportsFMX<V>`	Vista previa FMX (runtime / diseño).
`sgcReportsHTML<V>` / `dclsgcReportsHTML<V>`	Endpoint HTTP, visor de navegador y diseñador web (runtime / diseño).

Compílelos en orden de dependencia: primero los paquetes de runtime de sgcPDF, luego el núcleo de sgcReports, luego VCL o FMX, luego los paquetes de diseño y, por último, `sgcReportsHTML`. Después instale los paquetes de diseño `dcl*`. `TsgcReport` aparece en la página de paleta **sgcReports** después de instalar el paquete de diseño.

Ruta de búsqueda

Añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path), o añádala a la ruta de búsqueda de unidades del proyecto. sgcReports y sgcPDF comparten esta carpeta, así que una sola entrada cubre ambos.

C++Builder

Abra el proyecto `.cbproj` correspondiente a su versión de C++Builder en `sgcUI\Packages`, por ejemplo `sgcReportsC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

Primeros Pasos

Hay tres formas de construir un informe: la API de modelo (rellenar un componente TsgcReport con páginas, bandas y elementos), la API fluida (encadenar llamadas sobre un TsgcReportBuilder), o una definición de informe JSON cargada en tiempo de ejecución. Las tres terminan en el mismo TsgcReportEngine y las mismas exportaciones.

1. La API de modelo

Coloque un `TsgcReport`, un `TsgcReportEngine` y un `TsgcReportPDFExport` en un formulario (o créelos por código) y rellene el informe con una página, una banda y un elemento de texto:

```
var
  oPage: TsgcReportPage;
  oBand: TsgcReportBand;
  oPages: TsgcReportPreparedPages;
begin
  sgcReport.Units := ruPoints;
  with sgcReport.Styles.Add do
  begin
    Name := 'Title';
    Font.Name := 'Helvetica';
    Font.Size := 18;
    Font.Bold := True;
  end;

  oPage := sgcReport.Pages.Add;
  oPage.PaperSize := rpsA4;
  oPage.Margins.Left := 40;
  oPage.Margins.Top := 40;
  oPage.Margins.Right := 40;
  oPage.Margins.Bottom := 40;

  oBand := oPage.Bands.Add;
  oBand.BandType := rbtReportTitle;
  oBand.Height := 40;
  with oBand.Items.Add do
  begin
    Kind := rikText;
    Name := 'Title1';
    Text.Text := 'Hello, sgcReports';
    Width := 400;
    Height := 24;
    StyleName := 'Title';
  end;

  sgcEngine.Report := sgcReport;
  oPages := sgcEngine.Prepare;
  try
    sgcPDFExport.SaveToFile(oPages, 'hello.pdf');
  finally
    oPages.Free;
  end;
end;
```

2. La API fluida

El mismo informe con `TsgcReportBuilder`, que no necesita componentes en el formulario:

```
var
  vBuilder: TsgcReportBuilder;
begin
  vBuilder := TsgcReportBuilder.Create;
  try
    vBuilder.Units(ruPoints).Style('Title').Font('Helvetica', 18).Bold.EndStyle
      .Page.Paper(rpsA4).Margins(40, 40, 40, 40)
      .ReportTitle(40).Text('Hello, sgcReports', 0, 0, 400, 24).Style('Title')
      .EndBand.EndPage;
    vBuilder.SaveToPDF('hello.pdf');
  finally
    vBuilder.Free;
  end;
end;
```

Cada llamada a estilo, página y banda devuelve un subconstructor con ámbito propio (`TsgcReportStyleBuilder`, `TsgcReportPageBuilder`, `TsgcReportBandBuilder`); `EndStyle`, `EndBand` y `EndPage` regresan al ámbito contenedor para que la cadena pueda continuar.
Consulte TsgcReportBuilder para ver la cadena completa (estilos, páginas, bandas, elementos, datos, funciones auxiliares de exportación).

3. Una definición de informe JSON

Un informe construido de cualquiera de las dos formas se puede guardar como JSON y volver a cargar sin recompilar:

```
var
  vJSON: string;
  vLoaded: TsgcReport;
begin
  vJSON := sgcReportSaveToJSON(sgcReport);
  vLoaded := TsgcReport.Create(nil);
  try
    sgcReportLoadFromJSON(vLoaded, vJSON);
    // vLoaded now has the same pages, bands, items and styles as sgcReport
  finally
    vLoaded.Free;
  end;
end;
```

Consulte Definiciones de informes JSON para el formato del sobre, las reglas de nomenclatura y el esquema generado.

Próximos pasos

- Modelo de informe: páginas, bandas, elementos, estilos y unidades.
- Fuentes de datos: lee registros desde un `TDataSet`, JSON o eventos.
- Cómo hacer: factura con datos maestro-detalle.

Modelo de Informe

Un informe (TsgcReport) es un árbol de páginas, bandas, elementos y estilos, construido sobre `TCollection`, por lo que se serializa como un DFM igual que cualquier otro componente. El mismo árbol también se puede cargar y guardar como JSON, véase Definiciones de informes JSON.

Unidades

`TsgcReport.Units` (`TsgcReportUnits`: `ruMillimeters`, `ruPoints`, `ruInches`) es la unidad en la que se introduce cada `Left`, `Top`, `Width`, `Height` y margen. Internamente el diseño y la exportación a PDF siempre trabajan en puntos (1/72 de pulgada); el informe convierte al leer y escribir estas propiedades.

Páginas

`TsgcReportPage` (en `TsgcReport.Pages`) tiene `PaperSize` (`rpsA3`, `rpsA4`, `rpsA5`, `rpsLetter`, `rpsLegal`, `rpsCustom`), `Orientation`, `Width` y `Height` (usados solo cuando `PaperSize` es `rpsCustom`), `Margins`, `Columns` y `ColumnSpacing`. Un informe puede tener más de una definición de página; los subinformes `rbtDetailData` referencian una página por nombre (`TsgcReportItem.SubreportPageName`).

Bandas

`TsgcReportBand` (en `TsgcReportPage.Bands`) tiene un `BandType` que decide cuándo y con qué frecuencia se imprime:

BandType	Se imprime
rbtReportTitle	Una vez, antes del primer encabezado de página.
rbtReportSummary	Una vez, después del último grupo y datos de detalle.
rbtPageHeader	Parte superior de cada página.
rbtPageFooter	Parte inferior de cada página.
rbtColumnHeader	Parte superior de cada columna, en una página multicolumna.
rbtColumnFooter	Parte inferior de cada columna.
rbtGroupHeader	Una vez por cada valor distinto de `GroupExpression`, con `RepeatOnEveryPage` repitiéndola opcionalmente cuando un grupo abarca un salto de página.
rbtGroupFooter	Una vez al final de cada grupo, después de que sus agregados estén completos.
rbtMasterData	Itera los registros de `DataSourceName`; no imprime nada por sí misma.
rbtDetailData	Itera los registros de `DataSourceName` bajo el registro maestro actual.
rbtChild	Se imprime inmediatamente después de su banda padre a través de `ChildBandName`, compartiendo el mismo registro.

Otras propiedades de banda: `Height`, `CanGrow` y `CanShrink` (crecen o se reducen para ajustarse a su elemento más alto), `KeepTogether` (no divide la banda por un salto de página), `StartNewPage`, `PrintOnFirstPage` / `PrintOnLastPage`, `VisibleExpression` y `StyleName` (un estilo predeterminado para sus elementos).

Elementos y tipos

`TsgcReportItem` (en `TsgcReportBand.Items`) tiene un `Kind` (`TsgcReportItemKind`):

Kind	Subopciones	Notas
rikText	`Text` (`TsgcReportTextOptions`)	`Text.Text` es una plantilla de texto, véase Expresiones y plantillas de texto.
rikImage	`Image` (`TsgcReportImageOptions`)	Bytes de imagen incrustados, serializados como `PictureData`.
rikLine	—	Una línea horizontal o vertical.
rikShape	`Shape` (`TsgcReportShapeOptions`)	`ShapeKind`: rectángulo, rectángulo redondeado, elipse, línea.
rikBarcode	`Barcode` (`TsgcReportBarcodeOptions`)	11 simbologías, véase `TsgcReportBarcodeSymbology`; `Barcode.Value` es una plantilla de texto.
rikSubreport	`SubreportPageName`	Ejecuta en el sitio otra página del mismo informe.
rikPageBreak	—	Fuerza una nueva página.

Propiedades comunes de elemento: `Left`, `Top`, `Width`, `Height`, `StyleName`, `Visible` / `VisibleExpression`, `CanGrow` / `CanShrink`, `AltText` (accesibilidad) y `TagRole` (véase Accesibilidad).

Estilos

`TsgcReportStyle` (en `TsgcReport.Styles`) agrupa `Font`, `BackColor`, `Frame`, `Padding`, `HAlign` y `VAlign` bajo un `Name` que una banda o un elemento referencia mediante `StyleName`. Los colores son `TsgcReportColor`, un cardinal `$00RRGGBB` neutral respecto a la plataforma (no `Graphics.TColor`); `sgcReportColorNone`, `sgcReportColorBlack` y `sgcReportColorWhite` están predefinidos.

Expresiones y Plantillas de Texto

sgcReports compila cada expresión una vez y luego la evalúa una vez por registro. Los valores son Variants.

Campos, variables y ámbito

Un campo se escribe `Source.Field`, donde `Source` es el `Name` de una fuente de datos añadida al motor (`TsgcReportEngine.AddDataSource`) y `Field` es uno de sus nombres de campo. Un identificador simple se resuelve como una variable del motor (`TsgcReportEngine.Variables`, pares `Name=Value`), también accesible desde `OnGetVariable`. Siempre hay disponibles dos identificadores: `Page` (número de página actual) y `TotalPages`, que se resuelve solo después de la última página (diferido, en dos pasadas, "Página X de Y").
Un pie de grupo o un pie de página ve los valores de campo del **último** registro de su grupo o página, no del siguiente registro que provocó el salto.

Operadores

Aritméticos `+ - * /`, de comparación `= <> < <= > >=`, lógicos `and or not`, concatenación de cadenas con `+`, y paréntesis para agrupar.

Funciones

Función	Descripción
IIF(cond, a, b)	a cuando cond es verdadero; en caso contrario, b.
IsNull(v)	True cuando v es Null o está vacío.
Coalesce(v1, v2, ...)	Primer argumento que no es Null.
Upper(s)	Mayúsculas.
Lower(s)	Minúsculas.
Trim(s)	Elimina los espacios iniciales y finales.
Length(s)	Longitud de la cadena.
Copy(s, from, count)	Subcadena.
Pos(sub, s)	Posición base 1; 0 cuando no se encuentra.
Replace(s, from, to)	Sustituye todas las apariciones.
FormatFloat(fmt, v)	Formatea un número.
FormatDateTime(fmt, v)	Formatea una fecha u hora.
Str(v)	Convierte a cadena.
Round(v)	Redondea al entero más cercano.
Trunc(v)	Trunca hacia cero.
Abs(v)	Valor absoluto.

Int(v)	Parte entera.
MinOf(a, b)	El menor de los dos.
MaxOf(a, b)	El mayor de los dos.
Now()	Fecha y hora actuales.
Date()	Fecha actual.
Year(v)	Año de una fecha.
Month(v)	Mes de una fecha.
Day(v)	Día de una fecha.

Agregados

`Sum`, `Count`, `Avg`, `Min` y `Max` toman un argumento, por ejemplo `Sum(Lines.Qty * Lines.Price)`. Un agregado acumula a lo largo del ámbito en el que se imprime: dentro de un pie de grupo se reinicia en cada grupo, dentro de un pie de página se reinicia en cada página, dentro del resumen del informe se ejecuta para todo el informe.

Semántica de Null

Un campo ausente o una variable sin establecer se evalúa como Null, que se propaga a través de la aritmética (`Null + 1` es Null) y no se compara como igual ni como distinto de ningún valor no Null, siguiendo las reglas de Variant de Delphi. Use `IsNull` o `Coalesce` para sustituir un valor predeterminado antes de formatear.

Plantillas de texto

`TsgcReportTextTemplate` (usado por `TsgcReportItem.Text.Text` y `TsgcReportItem.Barcode.Value`) es texto literal con expresiones incrustadas entre corchetes, por ejemplo `'Invoice [Invoices.Number], total [Sum(Lines.Qty * Lines.Price)]'`. Todo lo que queda fuera de `[ ]` se copia tal cual, de modo que un valor de código de barras como `STOCK-2026-09` sin corchetes se imprime literalmente, ceros iniciales incluidos. `TsgcReportTextOptions.DisplayFormat` (o `Barcode.DisplayFormat` donde aplique) formatea el resultado evaluado de una plantilla de una sola expresión igual que lo harían `FormatFloat` o `FormatDateTime`.

Fuentes de datos

Una fuente de datos es un descendiente de `TsgcReportDataSource`, añadida a un TsgcReportEngine con `AddDataSource`. Su propiedad `SourceName` (no el `Name` del componente) es lo que las expresiones de banda referencian como `Source.Field`, y una banda `rbtMasterData` o `rbtDetailData` la referencia con el mismo nombre a través de `DataSourceName`.

Tipos

Clase	Lee de
TsgcReportDataSetSource	Cualquier `TDataSet` (VCL `TClientDataSet`, FireDAC, dbExpress, ...).
TsgcReportJSONSource	Un array JSON de objetos, desde una cadena, un archivo o un flujo.
TsgcReportEventSource	Controladores de eventos `OnFirst`, `OnNext`, `OnEof` y `OnGetValue`, para datos que no tienen forma de dataset ni de JSON, como una llamada REST paginada.

Maestro-detalle

`TsgcReportDataSource` (la base común) tiene `MasterSource`, `MasterFields` y `DetailFields` (separados por punto y coma, misma cantidad en ambos lados). Cuando una banda de detalle con `DataSourceName` apuntando a la fuente de detalle está anidada bajo una banda maestra, el motor reinicia la fuente de detalle y la ajusta al registro maestro actual según esos campos antes de cada iteración de detalle. Así es como la demo de facturas vincula las líneas de factura con su factura, véase Cómo hacer: factura con datos maestro-detalle.

Registros JSON

`TsgcReportJSONSource.RecordsPath` es una ruta con puntos hasta el array de registros; déjela vacía cuando la propia raíz JSON sea el array. Los nombres de campo se descubren a partir de los miembros del primer registro.

Seguridad en hilos

Una fuente de datos no se comparte entre informes preparados concurrentemente; cada hilo debe usar sus propias instancias de fuente (y su propio motor). Véase Seguridad en hilos y uso en servidor.

Grupos, Encabezados, Pies y Columnas

Grupos

Una banda `rbtGroupHeader` con una `GroupExpression` (por ejemplo `'Sales.Region'`) inicia un nuevo grupo cada vez que cambia el valor de esa expresión; su banda `rbtGroupFooter` correspondiente cierra el grupo una vez que sus agregados son definitivos. Establezca `RepeatOnEveryPage` en el encabezado para reimprimirlo cuando el grupo abarque un salto de página, y `StartNewPage` para forzar que cada grupo empiece en su propia página (usado para una página por factura). Los grupos se anidan: una página puede tener más de un par `rbtGroupHeader` / `rbtGroupFooter`, evaluados en el orden en que aparecen las bandas.

Encabezados y pies de página y de columna

`rbtPageHeader` y `rbtPageFooter` se imprimen en cada página. `rbtColumnHeader` y `rbtColumnFooter` se imprimen en cada columna cuando `TsgcReportPage.Columns` es mayor que 1. `PrintOnFirstPage` y `PrintOnLastPage` (predeterminado `True`) suprimen un encabezado o pie de página en la primera o última página.

Mantener unido

`TsgcReportBand.KeepTogether := True` obliga a que toda la banda quede en una sola página: si no cabe en el espacio restante de la página actual, el motor inicia una nueva página antes de imprimirla, en lugar de dividirla. Se usa en bloques de totales y otras bandas que no deben cortarse por un salto de página.

Columnas

`TsgcReportPage.Columns` (predeterminado 1) y `ColumnSpacing` disponen las bandas maestra y de detalle en columnas estilo periódico; el motor rellena una columna antes de pasar a la siguiente, imprimiendo `rbtColumnHeader` / `rbtColumnFooter` en cada límite de columna.

Subinformes

Un elemento `rikSubreport` (`TsgcReportItem.SubreportPageName`) ejecuta en el sitio otra `TsgcReportPage` del mismo informe, en la posición del elemento. La página del subinforme puede tener sus propias bandas y sus propias referencias de fuente de datos, y se prepara como parte de la misma ejecución.

Página X de Y

`Page` y `TotalPages` son los dos identificadores reservados para el texto de paginación, típicamente `'Page [Page] of [TotalPages]'` en una banda `rbtPageFooter`. `TotalPages` es diferido: el motor ejecuta todo el informe una vez para contar páginas y luego resuelve cada expresión diferida en una segunda pasada antes de entregar las páginas a una exportación.

Tipos de letra

`TsgcReportFontManager` (unidad `sgcReports_Fonts`) descubre, mide y aplica alternativas de tipos de letra tanto para el motor de diseño (ajuste de línea, `CanGrow`) como para la exportación a PDF. No tiene estado global; cada motor y exportación referencia un administrador a través de su propiedad `FontManager`.

Administrador predeterminado

Cuando `FontManager` se deja sin asignar, el motor y las exportaciones usan el `sgcReportDefaultFontManager` a nivel de proceso, una instancia compartida construida a partir de los tipos de letra instalados en el sistema, creada una vez en el primer uso. Asigne su propio `TsgcReportFontManager` cuando un informe necesite tipos de letra que el administrador predeterminado no tenga, o para mantener la carga de tipos de letra aislada entre informes concurrentes.

Descubrimiento

El administrador explora las carpetas de tipos de letra de la plataforma: la carpeta Fonts de Windows, los directorios estándar de tipos de letra de Linux, las ubicaciones de tipos de letra del sistema de macOS e iOS, y las rutas de tipos de letra y assets de Android. Las tipografías TrueType descubiertas se indexan por nombre de familia, grosor y estilo (negrita, cursiva).

Tipos de letra estándar y alternativa

Los 14 tipos de letra estándar de PDF (Helvetica, Times, Courier y sus variantes negrita y cursiva, más Symbol y ZapfDingbats) están mapeados de forma predeterminada; un estilo con `Font.Name := 'Helvetica'` se renderiza sin incrustación salvo que la exportación necesite una tipografía incrustable (PDF/A, o un carácter que el tipo de letra base no pueda representar). El administrador sustituye entonces una tipografía TrueType para la familia solicitada, y recurre aún más, glifo a glifo, a otra tipografía instalada cuando la actual no tiene un glifo para un carácter, lo que permite que un tipo de letra base latino aun así renderice texto CJK o cirílico presente en los datos.

Incrustación

La exportación a PDF incrusta las tipografías TrueType que realmente usó, reducidas a los glifos referenciados por el documento, salvo cuando PDF/A exige la incrustación completa de una tipografía ya parcialmente incrustada. `TsgcReportPDFExport.FontManager` resuelve las tipografías de los elementos sin tipografía explícita y, para la salida PDF/A, los sustitutos TrueType de los 14 tipos de letra estándar a través de `TsgcReportFontManager.ResolveEmbeddable`.

Plataformas móviles y sin interfaz

iOS no tiene una carpeta de tipos de letra del sistema legible desde una app en sandbox, y un daemon Linux64 puede ejecutarse en una imagen mínima con pocos tipos de letra instalados. En ambos casos, distribuya un tipo de letra TrueType (por ejemplo DejaVu Sans, de licencia libre de copyleft) junto al ejecutable o en el paquete de la app, y deje que el descubrimiento de tipos de letra recurra a él; véase Cómo hacer: un informe FMX móvil.

Seguridad en Hilos y Uso en Servidor

sgcReports está diseñado para la generación de informes en servidor y web, donde muchos informes se preparan y exportan al mismo tiempo en distintos hilos.

Regla

Un motor, su informe, sus fuentes de datos y sus exportaciones no se comparten entre hilos. Cada hilo de trabajo debe crear y usar su propio `TsgcReportEngine`, `TsgcReport` (o compartir una definición de informe de solo lectura y clonar únicamente el motor, ya que el motor mantiene el estado de la ejecución), sus propias instancias de fuente de datos y sus propios componentes de exportación. Nada en las clases de modelo, diseño o exportación usa un bloqueo global en torno al estado propio de un informe, de modo que dos hilos que comparten un motor o una exportación corrompen ambas ejecuciones.

Qué se puede compartir

Un `TsgcReportFontManager` puede compartirse en modo de solo lectura entre threads una vez descubiertas sus fuentes, incluido el `sgcReportDefaultFontManager` de ámbito de proceso: la detección y medición de fuentes no modifican un estado por llamada del que dependan otros threads. `TsgcReportFontManager.ResolveEmbeddable` tampoco cambia el ajuste `EmbedStandardFonts` del gestor, por lo que también es seguro llamarlo en un gestor compartido entre threads. Una `TsgcReportDefinition` (o las funciones simples `sgcReportLoadFromJSON` / `sgcReportSaveToJSON`) puede llamarse desde cualquier thread siempre que cada llamada trabaje con su propia instancia de `TsgcReport`.

Patrón: un grupo de trabajadores

Cargue una definición de informe JSON compartida y de solo lectura al arrancar. Cada hilo de trabajo hace entonces, para cada trabajo: crear un `TsgcReport`, llamar a `sgcReportLoadFromJSON` sobre él, crear un `TsgcReportEngine` y las fuentes de datos de ese trabajo, añadirlas, `Prepare`, exportar, y luego liberar todo lo que el hilo creó. Este es el diseño usado por la demo del daemon Linux, verificado con una prueba de estrés de 8 hilos en Win32, Win64 y Linux64.

Aplicaciones de servidor en Windows

`sgcReportsConfigureForServer` (unidad `sgcReports_Types`) es un ajuste opcional para servidores que preparan y exportan informes en muchos threads a la vez. En Windows, con Delphi 2006 o posterior, establece `System.NeverSleepOnMMThreadContention` en `True`, de modo que el gestor de memoria del RTL espera activamente en lugar de dormir cuando dos threads compiten por el mismo tamaño de bloque; no tiene efecto en otras plataformas ni en compiladores más antiguos. sgcReports nunca lo llama, por lo que no tiene efecto hasta que la propia aplicación lo llama, y una vez que lo hace, el ajuste se aplica a todo el proceso, a cada asignación de memoria que realice la aplicación, no solo a las exportaciones de informes.
Llámelo una sola vez, al iniciar la aplicación, antes de crear cualquier thread de trabajo, en un servidor multithread. Medido con 8 threads exportando 25 informes cada uno en Win64, el rendimiento subió de aproximadamente x1,8 a aproximadamente x5 respecto a un solo thread en dos ejecuciones (de 13,6 s a 4,4 s, y de 7,8 s a 2,8 s). El ajuste también hace que los threads usen más CPU mientras esperan, así que déjelo desactivado en aplicaciones de un solo thread.

```
program sgcReportsServerStartup;

{$APPTYPE CONSOLE}

uses
  sgcReports_Types;

begin
  // call once, at application start up, before any worker thread is created
```

```
  sgcReportsConfigureForServer;
end.
```

Endpoint HTTP

TsgcReportHTTPEndpoint registra un motor de larga duración por nombre de informe y sirve solicitudes concurrentes para él; véase el tema de ese componente para saber cómo sincroniza el acceso al motor compartido y a su caché de páginas preparadas.

Accesibilidad (PDF/UA)

TsgcReportPDFExport.PDFUA es `True` de forma predeterminada: todo PDF exportado se etiqueta para PDF/UA-1 (ISO 14289-1) salvo que se desactive explícitamente. El etiquetado escribe un árbol de estructura (encabezados, párrafos, tablas con filas de encabezado, figuras con texto alternativo, artefactos), el `/Lang` del documento, el título y el identificador `pdfuaid`, e incrusta todos los tipos de letra usados (los tipos de letra estándar se sustituyen por un sustituto TrueType a través del administrador de tipos de letra, ya que un PDF etiquetado no puede depender de una fuente base no incrustada).

Reglas de etiquetado automático

El rol de cada elemento en el árbol de estructura se deriva de su banda y su tipo, salvo que se sobrescriba:

Origen	Rol predeterminado
Elemento de texto en una banda `rbtReportTitle`	Encabezado (el nivel depende de `TagRole` si está establecido; en caso contrario, un encabezado de nivel superior).
Elemento de texto en una banda `rbtGroupHeader`	Encabezado.
Otros elementos de texto	Párrafo.
Una tabla construida a partir de filas de detalle repetidas bajo una banda de encabezado de columna	Tabla, con las celdas de la banda de encabezado de columna marcadas como celdas de encabezado de tabla.
Elementos de imagen y código de barras	Figura, usando `AltText` como texto alternativo.
Líneas y formas decorativas	Artefacto (excluido del orden de lectura presentado a la tecnología de asistencia).

TagRole

Establezca `TsgcReportItem.TagRole` para sobrescribir el rol automático: `rtrAuto` (predeterminado), `rtrHeading1`, `rtrHeading2`, `rtrHeading3`, `rtrParagraph`, `rtrFigure`, `rtrArtifact`, `rtrTableHeader` o `rtrTableCell`. Establezca siempre `AltText` en un elemento etiquetado `rtrFigure`, ya que el texto alternativo es lo que exige PDF/UA para que una figura sea válida.

Orden de lectura

El árbol de estructura sigue el orden en que se imprimen los elementos, de arriba abajo y de izquierda a derecha dentro de una banda, lo que en un diseño de informe normal ya coincide con el orden de lectura previsto; no existe un paso manual de ordenación aparte.

Verificación con veraPDF

Consulte Cómo hacer: un PDF accesible validado con veraPDF para conocer el comando que comprueba un archivo PDF/UA-1 frente al perfil ISO.

PDF/A

TsgcReportPDFExport.PDFA (`TsgcReportPDFAMode`) selecciona la salida de archivo: `rpaNone` (PDF simple predeterminado), `rpaA3b` (PDF/A-3b, reproducibilidad visual) o `rpaA3a` (PDF/A-3a, añade requisitos de etiquetado e idioma sobre A-3b). PDF/A-3 permite incrustar archivos arbitrarios en el documento, lo que lo convierte en la opción de archivo para la facturación electrónica y flujos de trabajo similares donde la fuente legible por máquina (por ejemplo el XML de la factura) viaja dentro del mismo PDF.

Combinación con PDF/UA

`PDFA = rpaA3a` y `PDFUA = True` se pueden establecer juntos: ambos requieren un árbol de estructura etiquetado, un idioma declarado y tipos de letra incrustados, de modo que la exportación produce un único árbol de estructura que satisface ambas declaraciones de conformidad. Esta es la combinación usada para facturas y otros documentos que deben ser accesibles y archivables a la vez; véase la guía de la factura y la guía de veraPDF.

Tipos de letra

PDF/A exige que se incruste todo tipo de letra. Cuando un estilo usa por nombre uno de los 14 tipos de letra estándar, la exportación sustituye e incrusta una tipografía TrueType a través del administrador de tipos de letra en lugar de depender del tipo de letra base, algo que PDF/A prohíbe como única representación del texto; véase Tipos de letra.

Salida reproducible

`FixedCreationDate` y `FixedDocumentID` en `TsgcReportPDFExport` fijan las fechas de creación y modificación y el identificador de archivo a un valor proporcionado por el llamador en lugar de la hora actual, de modo que exportaciones repetidas del mismo informe produzcan archivos idénticos byte a byte; útil para pruebas de archivo de referencia y deduplicación de archivo. Establecer cualquiera de los dos desactiva los flujos de objeto de PDF/A para que los valores sigan siendo parcheables.

TsgcReport

Unidad: sgcReports_Model
La definición del informe: páginas, bandas, elementos y estilos. Basado en `TComponent` y construido sobre `TCollection`, por lo que se serializa como un DFM igual que cualquier otro componente y también se puede cargar o guardar como JSON, véase Definiciones de informes JSON. Véase Modelo de informe para la descripción completa de páginas, bandas, elementos y estilos.

Propiedades

Propiedad	Tipo	Descripción
Unidades	TsgcReportUnits	Unidad en la que se introduce cada propiedad geométrica (Left, Top, Width, Height, márgenes): ruMillimeters, ruPoints o ruInches.
Páginas	TsgcReportPages	Colección de TsgcReportPage.
Estilos	TsgcReportStyles	Colección de TsgcReportStyle, referenciada mediante StyleName desde bandas y elementos.
Título	string	Título del informe; también el título predeterminado del documento PDF.
Autor	string	Autor del informe.

Ejemplo

Consulte Primeros pasos para ver un ejemplo completo de la API de modelo (estilos, una página, una banda, un elemento de texto) y Cómo hacer: factura con datos maestro-detalle para un informe completo.

TsgcReportEngine

Unidad: sgcReports_Engine
Ejecuta un TsgcReport contra sus fuentes de datos: recorre las relaciones maestro-detalle y los grupos, evalúa expresiones y agregados, y produce la lista de páginas preparadas que leen las exportaciones y los componentes de vista previa. Basado en `TComponent`.

Propiedades

Propiedad	Tipo	Descripción
Report	TsgcReport	La definición del informe a ejecutar.
Variables	TStrings	Parámetros de usuario como Name=Value, expuestos a las expresiones como identificadores simples.
FontManager	TsgcReportFontManager	No publicada (pública). Nil usa el sgcReportDefaultFontManager compartido. Véase Tipos de letra.
OnBeforePrintBand	TsgcReportBeforePrintBandEvent	Se dispara antes de que se imprima una banda; permite que el controlador cambie la visibilidad o los valores.
OnGetVariable	TsgcReportGetVariableEvent	Resuelve un identificador no encontrado en Variables.

Métodos

Método	Devuelve	Descripción
AddDataSource(aSource: TsgcReportDataSource)	—	Registra una fuente de datos bajo su SourceName; las expresiones de banda la referencian después como SourceName.Field.
RemoveDataSource(aSource: TsgcReportDataSource)	—	Anula el registro de una fuente de datos.
ClearDataSources	—	Elimina todas las fuentes de datos registradas.
Prepare	TsgcReportPreparedPages	Ejecuta el informe; quien realiza la llamada posee y libera el resultado.
Prepare(aPages: TsgcReportPreparedPages)	—	Sobrecarga que prepara sobre un TsgcReportPreparedPages existente, previamente vaciado.

Ejemplo

```
sgcEngine.Report := sgcReport;
sgcEngine.AddDataSource(srcInvoices);
sgcEngine.AddDataSource(srcLines);
oPages := sgcEngine.Prepare;
try
```

```
  pdfExport.SaveToFile(oPages, 'invoice.pdf');
finally
  oPages.Free;
end;
```

TsgcReportDataSetSource

Unidad: sgcReports_Data
Fuente de datos que lee registros de cualquier `TDataSet` (`TClientDataSet`, FireDAC, dbExpress, etc.). Desciende de `TsgcReportDataSource`, que proporciona las propiedades comunes maestro-detalle `MasterSource`, `MasterFields` y `DetailFields`, véase Fuentes de datos.

Propiedades

Propiedad	Tipo	Descripción
DataSet	TDataSet	El dataset a leer. Si se deja cerrado, la fuente lo abre y lo cierra en torno a cada ejecución; si ya está abierto, su estado y posición se restauran después.
SourceName	string	Heredado. El nombre que usan las expresiones de banda como Source.Field; independiente del Name del componente.
MasterSource	TsgcReportDataSource	Heredado. La fuente maestra de la que esta es un detalle.
MasterFields	string	Heredado. Nombres de campos maestros separados por punto y coma.
DetailFields	string	Heredado. Nombres de campos de detalle separados por punto y coma, en la misma cantidad que MasterFields.

Ejemplo

```
srcInvoices := TsgcReportDataSetSource.Create(Self);
srcInvoices.SourceName := 'Invoices';
srcInvoices.DataSet := cdsInvoices;

srcLines := TsgcReportDataSetSource.Create(Self);
srcLines.SourceName := 'Lines';
srcLines.DataSet := cdsLines;
srcLines.MasterSource := srcInvoices;
srcLines.MasterFields := 'Number';
srcLines.DetailFields := 'Invoice';
```

Consulte Cómo hacer: factura con datos maestro-detalle para ver el informe completo que usa este par de fuentes.

TsgcReportJSONSource

Unidad: sgcReports_Data
Fuente de datos que lee registros de un array JSON de objetos, dado como cadena, nombre de archivo o flujo. Desciende de `TsgcReportDataSource`, por lo que también admite `MasterSource`, `MasterFields` y `DetailFields`.

Propiedades

Propiedad	Tipo	Descripción
JSON	string	El texto JSON, cuando se establece directamente.
FileName	string	Un archivo del que leer el JSON.
RecordsPath	string	Ruta con puntos hasta el array de registros; vacío cuando la propia raíz JSON es el array.

Métodos

Método	Descripción
LoadFromStream(aStream: TStream)	Analiza el JSON UTF-8 del flujo de inmediato; sustituye a JSON y FileName hasta que alguno de los dos cambie.

Ejemplo

sales.json, un array bajo "sales":

```
{"sales": [
  {"region": "North", "customer": "Acme", "amount": 1200.50},
  {"region": "North", "customer": "Globex", "amount": 340.00}
]}
```

```
srcSales := TsgcReportJSONSource.Create(Self);
srcSales.SourceName := 'Sales';
srcSales.FileName := 'sales.json';
srcSales.RecordsPath := 'sales';
sgcEngine.AddDataSource(srcSales);
```

Consulte Cómo hacer: un informe agrupado con totales.

TsgcReportEventSource

Unidad: sgcReports_Data
Fuente de datos gobernada enteramente por eventos, para registros que no tienen forma `TDataSet` ni JSON, por ejemplo una respuesta REST paginada obtenida bajo demanda. Sin un controlador `OnEof`, la fuente no tiene registros.

Propiedades

Propiedad	Tipo	Descripción
FieldNames	TStrings	Los nombres de campo que expone esta fuente, ya que no hay dataset ni objeto JSON del que descubrirlos.
OnFirst	TNotifyEvent	Se dispara para reiniciar la iteración al primer registro.
OnNext	TNotifyEvent	Se dispara para avanzar al siguiente registro.
OnEof	TsgcReportEofEvent	`procedure(Sender: TObject; var aEof: Boolean) of object`. Establezca aEof en True cuando no queden más registros.
OnGetValue	TsgcReportGetValueEvent	`procedure(Sender: TObject; const aFieldName: string; var aValue: Variant) of object` . Devuelve el valor de un campo del registro actual.

Ejemplo

`OnFirst`, `OnNext`, `OnEof` y `OnGetValue` son eventos de puntero a método (declarados `of object`), por lo que reciben métodos con nombre de un formulario o una clase, no procedimientos anónimos. Este ejemplo usa dos campos, `FIndex` y `FRows`, declarados en el formulario contenedor junto con los cuatro métodos controladores siguientes.

```
procedure TForm1.DoFirst(Sender: TObject);
begin
  FIndex := -1;
end;

procedure TForm1.DoNext(Sender: TObject);
begin
  Inc(FIndex);
end;

procedure TForm1.DoEof(Sender: TObject; var aEof: Boolean);
begin
  aEof := FIndex >= High(FRows);
end;

procedure TForm1.DoGetValue(Sender: TObject; const aFieldName: string;
  var aValue: Variant);
begin
  if aFieldName = 'region' then aValue := FRows[FIndex][0]
  else if aFieldName = 'amount' then aValue := StrToFloat(FRows[FIndex][1]);
end;

procedure TForm1.SetupEventSource;
begin
```

```
  FRows[0] := ['North', '1200.5'];
  FRows[1] := ['South', '340'];
  FIndex := -1;

  srcEvt.FieldNames.CommaText := 'region,amount';
  srcEvt.OnFirst := DoFirst;
  srcEvt.OnNext := DoNext;
  srcEvt.OnEof := DoEof;
  srcEvt.OnGetValue := DoGetValue;
end;
```

`FRows` se declara como `array [0..1] of TArray<string>`. Este patrón funciona en todas las versiones de Delphi admitidas, de Delphi 7 a Delphi 13, ya que asigna punteros a método normales en lugar de métodos anónimos.

TsgcReportPDFExport

Unidad: sgcReports_Export_PDF
Renderiza las páginas preparadas a través de sgcPDF: tipos de letra TrueType incrustados y de subconjunto, tipos de letra base 14, texto de derecha a izquierda, rotación, recorte, texto justificado, formas, imágenes y códigos de barras vectoriales. Produce salida etiquetada PDF/UA-1 de forma predeterminada y salida de archivo PDF/A-3 opcional. Basado en `TComponent`.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Compress	Boolean	True	Flujos de objeto y una tabla de referencias cruzadas comprimida; los flujos de contenido siempre se comprimen.
PDFA	TsgcReportPDFAMode	rpaNone	rpaNone, rpaA3b o rpaA3a. Véase PDF/A.
PDFUA	Boolean	True	Escribe el árbol de estructura PDF/UA-1; se puede combinar con PDFA. Véase Accesibilidad.
Title / Author / Subject / Keywords	string	—	Metadatos del documento PDF.
Language	string	—	Idioma natural BCP 47, escrito por la exportación etiquetada.
Creator / Producer	string	—	Metadatos del documento PDF.
OnAfterRender	TsgcReportPDFDocumentEvent	—	Se dispara una vez que todas las páginas se han renderizado, antes de guardar el documento.
FontManager	TsgcReportFontManager	—	No publicada (pública). Véase Tipos de letra.
FixedCreationDate	TDateTime	0	No publicada (pública). Fija las fechas de creación y modificación de PDF/A para obtener una salida idéntica byte a byte.
FixedDocumentID	string	—	No publicada (pública). Fija el /ID al MD5 de este valor.

Métodos

Método	Descripción
SaveToStream(aPages: TsgcReportPreparedPages; aStream: TStream)	Renderiza las páginas preparadas en un flujo.

SaveToFile(aPages: TsgcReportPreparedPages; const aFileName: string)	Renderiza las páginas preparadas en un archivo.

Ejemplo: PDF accesible y de archivo

```
pdfExport.PDFA := rpaA3a;
pdfExport.PDFUA := True;
pdfExport.Language := 'en-US';
pdfExport.Title := sgcReport.Title;
oPages := sgcEngine.Prepare;
try
  pdfExport.SaveToFile(oPages, 'invoice_pdfa3a_ua1.pdf');
finally
  oPages.Free;
end;
```

TsgcReportXLSXExport

Unidad: sgcReports_Export_XLSX
Exporta las páginas preparadas como una hoja de cálculo de un libro .xlsx. Desciende de `TsgcReportGridExport`, que convierte las páginas preparadas en una cuadrícula (una fila por instancia de banda, columnas fusionadas a partir de las posiciones de los elementos dentro de una tolerancia) compartida con TsgcReportCSVExport. El paquete es un ZIP con entradas STORED (Content_Types, relaciones, libro, una hoja y estilos) que usa cadenas en línea, escrito sin ninguna librería de ZIP externa.

Propiedades (propias)

Propiedad	Tipo	Descripción
SheetName	string	Nombre de la hoja de cálculo.
FixedDateTime	TDateTime	Hora de modificación almacenada en las entradas del ZIP; 0 (predeterminado) usa la hora actual. Establézcala para obtener una salida idéntica byte a byte.

Propiedades (heredadas de TsgcReportGridExport)

Propiedad	Tipo	Predeterminado	Descripción
ExcludePageBands	Boolean	True	Excluye de la cuadrícula las bandas de encabezado de página, pie de página, encabezado de columna y pie de columna.
IncludeBandTypes	TsgcReportBandTypes	rbtReportTitle..rbtChild	Qué tipos de banda aportan filas.
PageFrom / PageTo	Integer	0	Intervalo de páginas (base 1); 0 significa todas las páginas.
DetectNumbers	Boolean	True	Las celdas de texto que se pueden interpretar como número se escriben como celdas numéricas.
ColumnTolerance	Double	—	Las posiciones Left más cercanas que este valor (puntos) se tratan como la misma columna.
DecimalSeparator / ThousandSeparator	Char	configuración regional	No publicada (pública). Usada por DetectNumbers.

Métodos (heredados)

Método	Descripción
SaveToStream(aPages, aStream)	Escribe el libro en un flujo.
SaveToFile(aPages, const aFileName)	Escribe el libro en un archivo.

BuildGrid(aPages): TsgcReportGrid	Construye la cuadrícula intermedia; quien realiza la llamada posee el resultado.

Ejemplo

```
xlsxExport.SheetName := 'Sales';
xlsxExport.SaveToFile(oPages, 'sales.xlsx');
```

TsgcReportCSVExport

Unidad: sgcReports_Export_CSV
Exporta las páginas preparadas como texto CSV, usando la misma cuadrícula basada en diseño que TsgcReportXLSXExport (véanse las propiedades de `TsgcReportGridExport` en esa página: ExcludePageBands, IncludeBandTypes, PageFrom, PageTo, DetectNumbers, ColumnTolerance).

Propiedades (propias)

Propiedad	Tipo	Predeterminado	Descripción
Delimiter	Char	','	Delimitador de campo.
QuoteChar	Char	'"'	Carácter de comillas; #0 desactiva el uso de comillas.
LineBreak	string	—	Separador de fila.
WriteBOM	Boolean	True	Escribe una marca de orden de bytes (BOM) UTF-8 al principio del archivo.

Métodos

Método	Devuelve	Descripción
GetCSV(aPages: TsgcReportPreparedPages)	string	Construye el texto CSV en memoria.
SaveToStream(aPages, aStream)	—	Escribe el CSV en un flujo.
SaveToFile(aPages, const aFileName)	—	Heredado de TsgcReportGridExport; escribe el CSV en un archivo.

Ejemplo

```
csvExport.Delimiter := ';';
csvExport.SaveToFile(oPages, 'sales.csv');
```

TsgcReportImageExport

Unidad: sgcReports_Export_Image
Exporta una imagen PNG por página, al DPI elegido, a través del renderizador ráster de sgcPDF, en todas las plataformas. Basado en `TComponent`.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
DPI	Integer	150	Píxeles por pulgada de las imágenes.
PageFrom / PageTo	Integer	0	Intervalo de páginas (base 1) para SaveToFiles; 0 significa la primera / última página.
BackgroundColor	TsgcReportColor	sgcReportColorWhite	Color de relleno antes de renderizar; sgcReportColorNone significa blanco.
FontManager	TsgcReportFontManager	—	No publicada (pública). Resuelve tipografías para elementos sin una tipografía explícita.

Métodos

Método	Devuelve	Descripción
PageCount(aPages: TsgcReportPreparedPages)	Integer	Número de páginas que escribiría SaveToFiles, aplicados PageFrom / PageTo.
SaveToStream(aPages, aPageIndex: Integer, aStream: TStream)	—	Escribe una página (base 0, se ignoran PageFrom / PageTo) como PNG.
SaveToFiles(aPages, const aFileNamePattern: string)	—	Escribe un archivo PNG por cada página del intervalo; aFileNamePattern contiene %d, sustituido por el número de página (base 1).
ClearCache	—	Descarta el PDF en caché usado internamente para rasterizar; la siguiente exportación lo regenera.

Ejemplo

```
imgExport.DPI := 150;
imgExport.SaveToFiles(oPages, 'sales_%d.png');
```

TsgcReportPreviewVCL

Unidad: sgcReports_Preview_VCL
Panel de vista previa VCL con navegación, zoom, impresión y exportación a PDF. Desciende de `TCustomPanel` y aloja internamente un `TsgcPDFViewerControl`, por lo que un informe siempre se renderiza a PDF para la vista previa.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Engine	TsgcReportEngine	—	El motor que ejecuta Prepare.
ShowToolbar	Boolean	True	Muestra la barra de herramientas integrada de navegación y zoom.
OnPageChange	TNotifyEvent	—	Se dispara cuando cambia la página mostrada.
PageCount	Integer	—	No publicada (pública), de solo lectura.
CurrentPage	Integer	—	No publicada (pública). Base 1; 0 cuando no se muestra nada.
Zoom	Double	—	No publicada (pública). Porcentaje; 100 es el tamaño real.
FontManager	TsgcReportFontManager	—	No publicada (pública). Véase LoadPrepared.
Viewer	TsgcPDFViewerControl	—	No publicada (pública), de solo lectura. El visor sgcPDF subyacente.

Métodos

Método	Descripción
Prepare	Ejecuta Engine.Prepare y muestra el resultado.
LoadPrepared(aPages: TsgcReportPreparedPages)	Muestra páginas preparadas en otro lugar, renderizadas con FontManager o, en su defecto, Engine.FontManager o, en su defecto, un administrador interno con los tipos de letra del sistema; pase el administrador con el que se prepararon las páginas.
Clear	Borra la vista previa.
FirstPage / PriorPage / NextPage / LastPage	Navegación.
ZoomIn / ZoomOut / FitWidth / FitPage	Control de zoom.
Print(const aTitle: string = '')	Imprime el documento mostrado.

SaveToPDF(const aFileName: string)	Escribe el PDF mostrado por la vista previa.

Ejemplo

```
sgcPreview.Engine := sgcEngine;
sgcPreview.Prepare;
lblStatus.Caption := Format('%d pages prepared', [sgcPreview.PageCount]);
```

TsgcReportPreviewFMX

Unidad: sgcReports_Preview_FMX
Vista previa FireMonkey para escritorio y móvil, réplica de TsgcReportPreviewVCL sobre `TControl` y un `TsgcPDFViewerFMX`. Se ejecuta en Windows, Android e iOS.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Engine	TsgcReportEngine	—	El motor que ejecuta Prepare.
ShowToolbar	Boolean	True	Muestra la barra de herramientas integrada de navegación y zoom.
OnPageChange	TNotifyEvent	—	Se dispara cuando cambia la página mostrada.
PageCount	Integer	—	No publicada (pública), de solo lectura.
CurrentPage	Integer	—	No publicada (pública). Base 1.
Zoom	Double	—	No publicada (pública). Porcentaje; 100 es el tamaño real.
Viewer	TsgcPDFViewerFMX	—	No publicada (pública), de solo lectura.

Los métodos y el resto de la superficie (Prepare, LoadPrepared, navegación, zoom, SaveToPDF) son los mismos que en TsgcReportPreviewVCL. Consulte Cómo hacer: un informe FMX móvil para un ejemplo completo, incluida la implementación de tipos de letra en iOS y Android.

TsgcReportHTTPEndpoint

Unidad: sgcReports_HTTP (paquete sgcReportsHTML)
Sirve informes registrados a través de un servidor HTTP de sgcWebSockets: páginas PDF, XLSX, CSV, PNG y un JSON informativo, con parámetros de consulta en lista blanca y una caché de páginas preparadas. Requiere la edición Professional de sgcWebSockets o superior. Basado en `TComponent`.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Server	TsgcHTTPServer_Base	—	El servidor HTTP al que engancharse. Opcional: en su lugar se puede llamar a HandleRequest desde su propio OnCommandGet.
BasePath	string	—	Prefijo de URL que posee el endpoint, por ejemplo /reports.
CacheSize	Integer	8	Conjuntos de parámetros preparados que se conservan por informe; 0 desactiva la caché.
MaxParamLength	Integer	256	Longitud máxima aceptada del valor de un parámetro de consulta.
DefaultDPI / MaxDPI	Integer	96 / 300	DPI usado y permitido para la ruta de página PNG.
OnBeforePrepare	TsgcReportHTTPBeforePrepareEvent	—	Se dispara antes de preparar un informe para una solicitud; puede validar o rechazar parámetros (por ejemplo, responder 403).
DesignerEnabled	Boolean	False	Activa las rutas del diseñador (schema, definition, preview) usadas por TsgcReportHTMLDesigner.
MaxDefinitionSize	Integer	1048576	Tamaño máximo aceptado del cuerpo de la definición de informe, en bytes.
OnBeforeSaveDefinition	TsgcReportHTTPBeforeSaveDefinitionEvent	—	Se dispara antes de guardar una definición enviada por el diseñador.

Métodos

Método	Devuelve	Descripción
RegisterReport(const aName: string; aEngine: TsgcReportEngine)	—	Registra un motor bajo un nombre; el motor no queda en propiedad. aName: letras, dígitos, '_' y '-'.
UnregisterReport(const aName: string)	—	Anula el registro de un informe.

FindEngine(const aName: string)	TsgcReportEngine	Busca un motor registrado.
GetReportParameters(const aName, aList: TStrings)	Boolean	Copia las Variables del motor (Name=Value) de un informe; False si no se conoce.
ClearCache(const aName: string = '')	—	Descarta las páginas preparadas en caché de un informe, o de todos los informes cuando aName está vacío.
GetReportPath(const aName: string)	string	Ruta URL de un informe sin extensión, por ejemplo /reports/sales.
HandleRequest(ARequestInfo, AResponseInfo)	Boolean	Sirve una solicitud que pertenece a BasePath y devuelve True; llámelo desde su propio OnCommandGet cuando Server no esté asignado.

Rutas

Para un informe registrado como "sales" bajo BasePath "/reports": `/reports/sales.pdf`, `/reports/sales.xlsx`, `/reports/sales.csv`, `/reports/sales/page/1.png?dpi=96` y `/reports/sales/info` (JSON: name, title, page count, parameters). Solo los parámetros de consulta cuyo nombre exista en las Variables del motor llegan al informe.

Rutas del diseñador

Disponibles solo cuando `DesignerEnabled` es `True` (en caso contrario, todas las rutas responden 404), y usadas por TsgcReportHTMLDesigner:

Método	Ruta	Descripción
GET	BasePath/schema.json	JSON Schema de las definiciones de informe, usado para gobernar el inspector de propiedades del diseñador.
GET	BasePath/{name}/definition	La definición de informe actual, en JSON.
PUT or POST	BasePath/{name}/definition	Valida el cuerpo, dispara `OnBeforeSaveDefinition` y sustituye el informe del motor registrado.
POST	BasePath/{name}/preview.png?page=1&dpi=96	Renderiza la definición del cuerpo de la solicitud sin sustituir el informe almacenado; la respuesta incluye la cabecera `X-Report-Page-Count`.

El cuerpo de la solicitud de las rutas de definición y vista previa debe ser `application/json` y no mayor que `MaxDefinitionSize`; una solicitud que falle en cualquiera de estas comprobaciones, o que falle la validación del esquema, se rechaza con un cuerpo de error JSON (`ok`, `path`, `message`), donde `path` es la ruta JSON del miembro problemático.
Consulte Cómo hacer: servir informes por HTTP para un servidor completo, y Cómo hacer: el diseñador de informes web para el propio diseñador.

TsgcReportHTMLViewer

Unidad: sgcReports_HTML_Viewer (paquete sgcReportsHTML)
Visor de navegador para un informe servido por un TsgcReportHTTPEndpoint: navegación de páginas, un formulario de parámetros y enlaces de descarga, incrustado en una página sgcHTML. Desciende de `TsgcHTMLComponent`.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Endpoint	TsgcReportHTTPEndpoint	—	El endpoint del que el visor lee las rutas del informe.
ReportName	string	—	Nombre del informe que se va a mostrar, tal como está registrado en el endpoint.
DPI	Integer	96	DPI usado para las imágenes de página.
ShowParameters	Boolean	—	Muestra un formulario de entrada para los parámetros del informe (Variables del motor).
ShowToolbar	Boolean	—	Muestra la navegación de páginas y los enlaces de descarga.

Ejemplo

```
reportViewer.Endpoint := reportEndpoint;
reportViewer.ReportName := 'sales';
reportViewer.ShowParameters := True;
```

Consulte Cómo hacer: servir informes por HTTP, que aloja el visor en "/" junto al endpoint en "/reports".

TsgcReportHTMLDesigner

Unidad: sgcReports_HTML_Designer (paquete sgcReportsHTML)
Diseñador de bandas basado en navegador que edita una definición de informe JSON conforme al esquema, mostrando una vista previa en vivo a través del mismo endpoint. Desciende de `TsgcHTMLComponent`. Requiere `TsgcReportHTTPEndpoint.DesignerEnabled := True`. Véase Cómo hacer: el diseñador de informes web.
La página que renderiza es un diseñador de bandas completo de arrastrar y soltar, no un simple editor de formularios:

- Un árbol de páginas, bandas y elementos a la izquierda, gobernado por la definición actual.
- Un lienzo de banda, en milímetros con independencia de las `Units` propias del informe, donde los elementos se arrastran y redimensionan con el puntero, se ajustan a la cuadrícula y se pueden desplazar con el teclado.
- Un inspector de propiedades generado a partir de `schema.json`, de modo que sigue los campos y enumeraciones de la definición del informe en lugar de un formulario fijo.
- Guardar, que envía (POST) la definición a `{name}/definition` y resalta el campo problemático cuando el servidor la rechaza, a partir de la ruta JSON de la respuesta de error.
- Vista previa, que envía (POST) la definición a `{name}/preview.png` y muestra la página renderizada sin guardarla.
- Deshacer y rehacer sobre el historial de edición en memoria, y un botón Descargar JSON que guarda la definición actual como archivo.

Propiedades

Propiedad	Tipo	Descripción
Endpoint	TsgcReportHTTPEndpoint	El endpoint del que el diseñador lee las rutas de esquema, definición y vista previa; debe tener `DesignerEnabled` establecido en `True`.
ReportName	string	Nombre de la definición de informe que se está editando.
DesignerHeight	Integer	Altura, en píxeles CSS, del diseñador incrustado; predeterminado 640.

Ejemplo

```
reportEndpoint.DesignerEnabled := True;
reportDesigner.Endpoint := reportEndpoint;
reportDesigner.ReportName := 'sales';
```

TsgcReportBuilder

Unidad: sgcReports_Fluent
API fluida code-first: estilos, páginas, bandas y elementos construidos con llamadas encadenadas que devuelven `Self` o un subconstructor con ámbito propio, de modo que todo el informe es un único árbol de expresiones. Funciona en Delphi 7 (una clase sencilla, sin necesidad de métodos anónimos). Construye y posee su propio `TsgcReport` y `TsgcReportEngine`, salvo que se pase un informe existente al constructor.

Constructores

Constructor	Descripción
Create	Crea y posee un nuevo TsgcReport.
Create(aReport: TsgcReport)	Construye sobre un informe existente, sin poseerlo.

Métodos

Método	Devuelve	Descripción
Title / Author / Subject / Keywords / Language(const aValue: string)	TsgcReportBuilder	Establece la propiedad correspondiente de TsgcReport.
Units(aUnits: TsgcReportUnits)	TsgcReportBuilder	Establece TsgcReport.Units.
Style(const aName: string)	TsgcReportStyleBuilder	Busca o añade el estilo aName; encadene Font, Bold, Italic, Underline, Color, BackColor, Frame, Padding sobre el resultado.
Page(const aName: string = '')	TsgcReportPageBuilder	Añade una página en cada llamada; encadene llamadas a PaperSize, Margins, Band y elementos Text/Barcode/Image/Line.
Data(aSource: TsgcReportDataSource)	TsgcReportBuilder	Añade una fuente de datos al motor; sin poseerla.
Variable(const aName, aValue: string)	TsgcReportBuilder	Establece una variable del motor.
FontManager(aManager: TsgcReportFontManager)	TsgcReportBuilder	Asigna el administrador de tipos de letra del motor; sin poseerlo.
Flush	—	Escribe los modificadores pendientes de la banda o elemento actual.
Prepare	TsgcReportPreparedPages	Prepara el informe; quien realiza la llamada posee el resultado.
Prepare(aPages: TsgcReportPreparedPages)	—	Sobrecarga que prepara sobre un objeto de páginas existente.
SaveToPDF(const aFileName) / SaveToPDFStream(aStream)	—	Prepara y exporta a través de PDFExport; Title, Author, Subject, Keywords y Language del infor-

		me se copian en PDFExport antes de cada guardado.
SaveToXLSX / SaveToXLSXStream	—	Prepara y exporta a XLSX.
SaveToCSV / SaveToCSVStream	—	Prepara y exporta a CSV.
SaveToPNG(const aFileNamePattern: string; aDPI: Integer = 150)	—	Un archivo PNG por página; aFileNamePattern contiene %d.

Propiedades

Propiedad	Tipo	Descripción
Report	TsgcReport	El informe, con los modificadores pendientes ya escritos.
Engine	TsgcReportEngine	El motor del constructor, conectable a un TsgcReportPreviewVCL / TsgcReportPreviewFMX.
PDFExport	TsgcReportPDFExport	Se crea en el primer uso, propiedad del constructor; establezca PDFA, PDFUA o Compress antes de guardar.

Ejemplo

Consulte Primeros pasos para ver un ejemplo mínimo, y Cómo hacer: un informe agrupado con totales para un informe completo construido enteramente con la API fluida.

Definiciones de informes JSON

Unidad: sgcReports_Definition
Carga y guarda un TsgcReport completo como JSON, de modo que un informe creado una vez con el modelo o la API fluida se puede editar, almacenar y volver a implementar sin necesidad de recompilar.

Sobre

```
{"format": "sgcReports", "version": 1, "report": { ... }}
```

Se acepta un miembro de cadena opcional `"$schema"` en el sobre.

Convenciones

- Cada propiedad publicada del modelo se escribe con su nombre de Delphi en lowerCamelCase (`DataSourceName` pasa a `dataSourceName`, `HAlign` pasa a `hAlign`); el `Name` del componente del informe no forma parte de la definición.
- Las enumeraciones se escriben como sus identificadores de Delphi (`"rbtMasterData"`), los conjuntos como arrays de identificadores, los booleanos como `true` / `false`, y los enteros y flotantes como números JSON; los colores son enteros `$00RRGGBB`.
- Los subobjetos `TPersistent` son objetos anidados, las colecciones son arrays de objetos de elemento y los `TStrings` son arrays de cadenas.
- Los valores iguales a su valor predeterminado se omiten, igual que en un DFM.
- Los bytes de imagen incrustados de un elemento de imagen se escriben como la cadena base64 `pictureData` de su objeto `image`.

La carga borra primero el informe. Cualquier error genera `ESgcReportDefinitionError` con la ruta JSON del miembro problemático, por ejemplo `report.pages[0].bands[2].height`. Un miembro desconocido es un error a menos que `IgnoreUnknown` sea `True`. Los flujos y archivos son UTF-8 sin BOM al guardar; se tolera un BOM al cargar.

Funciones

Función	Descripción
sgcReportSaveToJSON(aReport: TsgcReport; aIndent: Boolean = True): string	Serializa un informe a una cadena JSON.
sgcReportLoadFromJSON(aReport: TsgcReport; const aJSON: string)	Carga un informe desde una cadena JSON.
sgcReportSaveToJSONFile / sgcReportLoadFromJSONFile	Variantes de archivo.
sgcReportSaveToJSONStream / sgcReportLoadFromJSONStream	Variantes de flujo.
sgcReportGenerateSchema: string	JSON Schema (draft 2020-12) de la definición, generado a partir de RTTI; se usa como sgcReports.schema.json.

TsgcReportDefinition

La clase detrás de las funciones simples, para quienes necesiten controlar `IgnoreUnknown` o `Indent`: `SaveToJSON`, `LoadFromJSON`, `SaveToStream` / `LoadFromStream`, `SaveToFile` / `LoadFromFile`, y las propiedades `IgnoreUnknown` (predeterminado False) e `Indent` (predeterminado True, sangría de dos espacios y saltos de línea LF).

Ejemplo

Consulte Primeros pasos y Cómo hacer: el diseñador de informes web, que edita un definition.json y un data.json.

Cómo Hacer: Factura con Datos Maestro-Detalle

Esto recorre la demo `ReportInvoice` (`delphi\Demos\Delphi\ReportInvoice`): dos `TClientDataSet` (facturas y líneas de factura) leídos a través de dos TsgcReportDataSetSource, enlazados mediante `MasterSource`, impresos como una página por factura con un encabezado de columna repetido, subtotal, IVA y total, un código QR con el número de factura, y Página X de Y. El informe se muestra en un TsgcReportPreviewVCL y se exporta como un archivo PDF/A-3a que también es PDF/UA-1.

Vincular las dos fuentes

```
srcInvoices.SourceName := 'Invoices';
srcInvoices.DataSet := cdsInvoices;

srcLines.SourceName := 'Lines';
srcLines.DataSet := cdsLines;
srcLines.MasterSource := srcInvoices;
srcLines.MasterFields := 'Number';
srcLines.DetailFields := 'Invoice';

sgcEngine.AddDataSource(srcInvoices);
sgcEngine.AddDataSource(srcLines);
```

Una página por factura

El encabezado de grupo que inicia cada factura establece `StartNewPage`, y su `GroupExpression` es el número de factura, de modo que un nuevo grupo (y una nueva página) comienza cada vez que este cambia:

```
oBand := oPage.Bands.Add;
oBand.BandType := rbtGroupHeader;
oBand.Name := 'InvoiceHeader';
oBand.Height := 110;
oBand.GroupExpression := 'Invoices.Number';
oBand.StartNewPage := True;
```

El elemento de encabezado dentro de él se etiqueta como `rtrHeading1` para PDF/UA, y el valor del código QR es una plantilla de texto con el número de factura, no una expresión por sí sola, de modo que se conservaría el texto literal circundante:

```
oItem := oBand.Items.Add;
oItem.Kind := rikText;
oItem.Text.Text := 'INVOICE';
oItem.TagRole := rtrHeading1;
...
oItem := oBand.Items.Add;
oItem.Kind := rikBarcode;
oItem.Barcode.Symbology := rbsQRCode;
oItem.Barcode.Value := '[Invoices.Number]';
oItem.AltText := 'QR code with the invoice number';
```

La tabla de líneas y su encabezado repetido

El encabezado de columna de la tabla de líneas es una segunda banda `rbtGroupHeader` sobre la misma `GroupExpression`, con `RepeatOnEveryPage := True` para que se reimprima cuando las líneas de una factura larga continúen en una segunda página. La banda maestra itera las facturas y no imprime nada; la banda de detalle itera las líneas:

```
oBand := oPage.Bands.Add; // LinesHeader
oBand.BandType := rbtGroupHeader;
oBand.GroupExpression := 'Invoices.Number';
oBand.RepeatOnEveryPage := True;

oBand := oPage.Bands.Add; // InvoiceData (master), prints nothing
oBand.BandType := rbtMasterData;
oBand.DataSourceName := 'Invoices';

oBand := oPage.Bands.Add; // LineData (detail)
oBand.BandType := rbtDetailData;
oBand.DataSourceName := 'Lines';
```

Totales e IVA

El pie de grupo ve los campos de la última línea impresa y de la propia factura, de modo que su tipo de IVA proviene directamente de `Invoices.VATRate` aunque el último registro iterado fuera una línea:

```
// Subtotal
'[Sum(Lines.Qty * Lines.Price)]'
// VAT, using the invoice's own field from the group footer
'VAT [Invoices.VATRate] %'
'[Sum(Lines.Qty * Lines.Price) * Invoices.VATRate / 100]'
// Total
'[Sum(Lines.Qty * Lines.Price) * (1 + Invoices.VATRate / 100)]'
```

PDF accesible y de archivo

```
// PDFA and PDFUA are set on pdfExport in the form
oPages := sgcEngine.Prepare;
try
  pdfExport.SaveToFile(oPages, 'invoice_pdfa3a_ua1.pdf');
finally
  oPages.Free;
end;
```

El proyecto completo y compilable es `delphi\Demos\Delphi\ReportInvoice\sgcReports_Demo_Invoice.dpr`.

Cómo Hacer: Un Informe Agrupado con Totales

Esto recorre la demo `ReportFluent` (`delphi\Demos\Delphi\ReportFluent`): una lista de ventas leída de `sales.json`, agrupada por región, con un subtotal por región y un total general, construida enteramente con TsgcReportBuilder. Un equivalente con fuente JSON y API de modelo, agrupado por región y cliente, es la demo `ReportGroups`.

Fuente de datos y estilos

```
FSales := TsgcReportJSONSource.Create(Self);
FSales.SourceName := 'Sales';
FSales.FileName := ExtractFilePath(ParamStr(0)) + 'sales.json';

FBuilder := TsgcReportBuilder.Create;
FBuilder.Title('Sales by region').Author('eSeGeCe').Units(ruMillimeters)
  .Style('Base').Font('Arial', 10).EndStyle
  .Style('Title').Font('Arial', 18).Bold.EndStyle
  .Style('Group').Font('Arial', 11).Bold.BackColor($EEEEEE).EndStyle
```

Página, título y encabezados de columna

```
  .Page.Paper(rpsA4).Margins(15)
  .ReportTitle(14).Style('Title').Text('Sales by region', 0, 0, 180, 10)
    .TagRole(rtrHeading1).EndBand
  .PageHeader(8).Style('Base').Bold.Text('Customer', 2, 0, 68, 6)
    .Text('Product', 70, 0, 60, 6).Text('Amount', 130, 0, 50, 6)
    .AlignRight.EndBand
```

Agrupar por región, una línea por venta, subtotal

```
const
  CS_TOTAL = '[FormatFloat(''#,0.00'', Sum(Sales.Amount))]';
begin
  ...
  .GroupHeader('Sales.Region', 9).Style('Group')
    .Text('Region: [Sales.Region]', 0, 1, 180, 7).TagRole(rtrHeading2).EndBand
  .MasterData('Sales', 6).Style('Base').Text('[Sales.Customer]', 2, 0, 68, 6)
    .Text('[Sales.Product]', 70, 0, 60, 6).Text('[Sales.Amount]', 130, 0, 50, 6)
    .AlignRight.Format('#,0.00').EndBand
  .GroupFooter(9).Style('Base')
    .Text('Region total ' + CS_TOTAL, 80, 1, 100, 6).AlignRight.Bold.EndBand
```

`GroupHeader` acepta tanto una expresión simple (`'Sales.Region'`) como una envuelta entre `[ ]`. El agregado `Sum` en un pie de grupo se reinicia en cada región; la misma expresión en la banda de resumen inferior se ejecuta para todo el informe.

Total general y números de página

```
  .Summary(12).Style('Base').Line(0, 2, 180, 2)
    .Text('Grand total ' + CS_TOTAL, 80, 4, 100, 7).AlignRight.Bold.EndBand
  .PageFooter(8).Style('Base').Text('Page [Page] of [TotalPages]', 0, 1, 180, 6)
    .AlignCenter.EndBand
  .EndPage.Data(FSales);
```

Vista previa y exportación

```
FPreview.Engine := FBuilder.Engine;
FBuilder.Flush;
FPreview.Prepare;
...
FBuilder.SaveToPDF(OutputFile('sales.pdf'));
FBuilder.SaveToXLSX(OutputFile('sales.xlsx'));
FBuilder.SaveToCSV(OutputFile('sales.csv'));
FBuilder.SaveToPNG(OutputFile('sales_%d.png'), 96);
```

El proyecto completo y compilable es `delphi\Demos\Delphi\ReportFluent\sgcReports_Demo_Fluent.dpr`; el mismo patrón de agrupación y totales con una fuente `TDataSet` o de API de modelo también se muestra en `ReportGroups`.

Cómo Hacer: Exportar a XLSX, CSV y PNG

Esto recorre la demo `ReportExports` (`delphi\Demos\Delphi\ReportExports`): un informe se prepara una vez y luego se exporta a través de cuatro componentes distintos. Las páginas preparadas (`TsgcReportPreparedPages`) se reutilizan en cada exportación, de modo que el diseño y el recorrido de los datos solo se ejecutan una vez.

```
sgcEngine.Report := sgcReport;
sgcEngine.AddDataSource(srcOrders);
FPages := sgcEngine.Prepare;
sgcPreview.LoadPrepared(FPages);
```

PDF

```
pdfExport.SaveToFile(FPages, OutputFolder + 'orders.pdf');
```

XLSX

TsgcReportXLSXExport convierte las páginas preparadas en una cuadrícula (una fila de hoja de cálculo por instancia de banda) antes de escribir el libro:

```
xlsxExport.SheetName := 'Orders';
xlsxExport.SaveToFile(FPages, OutputFolder + 'orders.xlsx');
```

CSV

```
csvExport.Delimiter := ',';
csvExport.SaveToFile(FPages, OutputFolder + 'orders.csv');
```

PNG, un archivo por página

```
imageExport.DPI := 150;
imageExport.SaveToFiles(FPages, OutputFolder + 'orders_%d.png');
```

Notas

- De forma predeterminada (`ExcludePageBands = True`) los encabezados y pies de página y de columna no se convierten en filas de cuadrícula en la exportación XLSX y CSV; solo lo hacen el título del informe, y las bandas de grupo y de datos.
- `DetectNumbers` (predeterminado `True`) escribe una celda de texto que se interpreta como número como una celda numérica de XLSX en lugar de como texto.
- `SaveToFiles` nombra los archivos sustituyendo `%d` en el patrón por el número de página (base 1), de modo que `orders_%d.png` produce `orders_1.png`, `orders_2.png`, y así sucesivamente.

El proyecto completo y compilable es `delphi\Demos\Delphi\ReportExports\sgcReports_Demo_Exports.dpr`.

Cómo Hacer: Un PDF Accesible Validado con veraPDF

Esto recorre la demo `ReportPDFUA` (`delphi\Demos\Delphi\ReportPDFUA`): un título H1, encabezados de grupo H2, una tabla con filas de encabezado, una imagen con texto alternativo y un código de barras, exportados como PDF/UA-1, luego como PDF/A-3b, y luego como PDF/A-3a y PDF/UA-1 juntos. Véase Accesibilidad y PDF/A para las reglas de etiquetado y los niveles de conformidad.

Etiquetar el contenido

```
// title
oItem.TagRole := rtrHeading1;
// group heading
oItem.TagRole := rtrHeading2;
// logo, needs alternate text to satisfy PDF/UA
oItem.Kind := rikImage;
oItem.Image.FileName := 'logo.png';
oItem.AltText := 'eSeGeCe logo';
// decorative rule: excluded from the reading order
oItem.Kind := rikLine;
oItem.TagRole := rtrArtifact;
```

Una tabla construida a partir de una banda de encabezado de grupo o de columna seguida de filas de detalle se etiqueta automáticamente; no se necesita nada más para que sus celdas de encabezado queden marcadas como `rtrTableHeader`.

Exportar las tres variantes

```
// PDF/UA-1 only
pdfExport.PDFA := rpaNone;
pdfExport.PDFUA := True;
pdfExport.Language := 'en-US';
pdfExport.SaveToFile(oPages, 'report_ua1.pdf');

// PDF/A-3b only (visual reproducibility, no tagging required)
pdfExport.PDFA := rpaA3b;
pdfExport.PDFUA := False;
pdfExport.SaveToFile(oPages, 'report_a3b.pdf');

// both: PDF/A-3a and PDF/UA-1
pdfExport.PDFA := rpaA3a;
pdfExport.PDFUA := True;
pdfExport.SaveToFile(oPages, 'report_a3a_ua1.pdf');
```

Validar con veraPDF

veraPDF es un validador de código abierto de PDF/A y PDF/UA respaldado por ISO, que se ejecuta por separado de sgcReports como herramienta de línea de comandos. Compruebe cada archivo frente a la variante correspondiente:

```
verapdf --flavour ua1 report_ua1.pdf
verapdf --flavour 3b  report_a3b.pdf
verapdf --flavour 3a  report_a3a_ua1.pdf
verapdf --flavour ua1 report_a3a_ua1.pdf
```

Una ejecución correcta informa `compliant: true` para la variante solicitada; un fallo enumera la cláusula ISO y el objeto que la incumple, que es la forma más rápida de encontrar un `AltText` ausente o un elemento que debería haberse etiquetado como `rtrArtifact`.
El proyecto completo y compilable es `delphi\Demos\Delphi\ReportPDFUA\sgcReports_Demo_PDFUA.dpr`.

Cómo Hacer: Servir Informes por HTTP

Esto recorre la demo `ReportHTTPServer` (`delphi\Demos\Delphi\ReportHTTPServer`): un `TsgcWebSocketHTTPServer` en el puerto 8080 con un TsgcReportHTTPEndpoint sirviendo un informe "sales", y una página sgcHTML en `/` que aloja un TsgcReportHTMLViewer. Necesita el paquete `sgcReportsHTML`: sgcWebSockets (edición Professional o superior), sgcHTML y sgcReports.

Registrar el informe y engancharlo al servidor

```
reportEndpoint.BasePath := '/reports';
reportEndpoint.Server := httpServer;
reportEndpoint.RegisterReport('sales', salesEngine);

httpServer.Port := 8080;
httpServer.Active := True;
```

Rutas servidas

- `/reports/sales.pdf`, `/reports/sales.xlsx`, `/reports/sales.csv`
- `/reports/sales/page/1.png?dpi=96`
- `/reports/sales/info` (JSON: name, title, page count, parameters)

Restringir parámetros

El endpoint acepta solo los parámetros de consulta cuyo nombre exista en las `Variables` del motor de informes, de modo que un informe con una variable llamada `Region` acepta solo `?Region=...`. `OnBeforePrepare` puede validar el valor más a fondo, por ejemplo rechazando cualquier valor fuera de una lista permitida antes de preparar el informe:

```
salesEngine.Variables.Values['Region'] := 'All';
// OnBeforePrepare checks the incoming Region against an allowed list
// (All, North, South, East, West) and can reject the request (403)
// or load the rows for that region before the report is prepared.
```

Servir la página del visor

```
reportViewer.Endpoint := reportEndpoint;
reportViewer.ReportName := 'sales';
reportViewer.ShowParameters := True;
```

Notas de compilación

Compile con la carpeta `Source` de sgcWebSockets en la ruta de búsqueda. Defina `SGC_CUSTOM_INDY` al enlazar contra los paquetes publicados de sgcWebSockets (construidos sobre el fork sgcIndy), o `SGC_REPORTS_STOCK_INDY` cuando sgcWebSockets se construyó contra el Indy que incluye el IDE.
El proyecto completo y compilable es `delphi\Demos\Delphi\ReportHTTPServer\sgcReports_Demo_HTTPServer.dpr`.

Cómo Hacer: Un Daemon de Informes Linux64

Esto recorre la demo `ReportLinuxDaemon` (`delphi\Demos\Delphi\ReportLinuxDaemon`), una aplicación de consola que se ejecuta sin cambios en Windows y Linux64, ya que el núcleo de sgcReports no tiene dependencia de VCL ni de FMX. Véase Seguridad en hilos y uso en servidor para el patrón de grupo de trabajadores que usa.

Archivos de trabajo

El daemon vigila una carpeta de trabajos en busca de archivos de trabajo `*.json`. Un trabajo indica una definición de informe, un archivo de datos y un formato de salida (`pdf`, `xlsx`, `csv` o `png`); las rutas relativas del trabajo se resuelven respecto a la propia carpeta del archivo de trabajo. Un trabajo se reclama renombrándolo a `.json.working`, la salida se escribe junto a él, y el trabajo termina como `.json.done` o `.json.failed`.

Grupo de trabajadores

Un grupo de 4 hilos de trabajo renderiza los trabajos de forma concurrente. Cada hilo carga la definición de informe del trabajo con `sgcReportLoadFromJSON` en su propio `TsgcReport`, crea su propio `TsgcReportEngine` y fuente de datos, prepara, exporta y luego libera todo lo que creó, siguiendo el patrón compartido y de solo lectura descrito en Seguridad en hilos y uso en servidor.

Ejecutarlo

```
sgcReports_Demo_LinuxDaemon JOBS_FOLDER
sgcReports_Demo_LinuxDaemon --sample FOLDER   -- writes sample.job plus its report definition and data
```

SIGTERM o SIGINT en Linux, y Ctrl+C en Windows, detienen el daemon de forma ordenada: no se reclama ningún trabajo nuevo, los trabajos en curso terminan y el proceso finaliza con código 0.

Unidad systemd

`sgcreports-daemon.service` ejecuta el ejecutable compilado bajo un usuario dedicado, vigila `/var/lib/sgcreports/jobs`, se reinicia en caso de fallo y se detiene con SIGTERM con un tiempo de espera de 60 segundos para que terminen los trabajos en curso. Los registros van a stdout, léalos con:

```
journalctl -u sgcreports-daemon -f
```

El proyecto completo y compilable es `delphi\Demos\Delphi\ReportLinuxDaemon\sgcReports_Demo_LinuxDaemon.dpr`.

Cómo Hacer: Un Informe FMX Móvil

Esto recorre la demo `ReportMobileFMX` (`delphi\Demos\Delphi\ReportMobileFMX`), un pequeño albarán A5 (un código QR con el número de albarán, líneas de artículos, unidades totales y una línea de firma) construido con la API fluida y mostrado en un TsgcReportPreviewFMX. Compila para Win32, Win64, Android64 e iOS desde un único proyecto.

Ubicación de guardado

El botón Save PDF escribe `sgcReports_DeliveryNote.pdf` en la carpeta de documentos de la app y muestra la ruta:

```
uses System.IOUtils;
...
vBuilder.SaveToPDF(TPath.Combine(TPath.GetDocumentsPath,
  'sgcReports_DeliveryNote.pdf'));
```

Tipos de letra en móvil

La demo usa los tipos de letra ya presentes en el sistema mediante el descubrimiento de tipos de letra (véase Tipos de letra). iOS no tiene una carpeta de tipos de letra del sistema legible desde una app en sandbox, de modo que, cuando no se encuentra un tipo de letra TrueType adecuado, la demo carga en su lugar `DejaVuSans.ttf` desde su propia carpeta; `DejaVu-LICENSE.txt` es la licencia de ese tipo de letra y se distribuye con él.

Implementar el tipo de letra

En el IDE, Project > Deployment, añada `DejaVuSans.ttf` para cada plataforma móvil con estas rutas remotas:

Plataforma	Ruta remota
Android64	assets\internal\
iOS	StartUp\Documents\

En Windows, el tipo de letra se lee de la carpeta del ejecutable, así que cópielo allí cuando el sistema no tenga instalado un tipo de letra adecuado.
El proyecto completo y compilable es `delphi\Demos\Delphi\ReportMobileFMX\sgcReports_Demo_MobileFMX.dpr`; requiere sgcReports, sgcPDF y FireMonkey.

Cómo Hacer: El Diseñador de Informes Web

TsgcReportHTMLDesigner es un diseñador de bandas basado en navegador que edita una definición de informe JSON conforme a `sgcReports.schema.json`, con una vista previa en vivo servida por el mismo TsgcReportHTTPEndpoint que un visor o una aplicación usa para servir el informe terminado.

Activarlo

```
reportEndpoint.DesignerEnabled := True;
reportDesigner.Endpoint := reportEndpoint;
reportDesigner.ReportName := 'sales';
```

`DesignerEnabled` activa las rutas de schema, definition y preview que llama la página del diseñador; déjelo en `False` en un endpoint de producción que solo sirve informes terminados, y actívelo únicamente donde se pretenda editar, protegido por su propia autenticación.

Qué ofrece el diseñador en el navegador

La página servida por `TsgcReportHTMLDesigner` es un diseñador de bandas completo de arrastrar y soltar: un árbol de páginas, bandas y elementos; un lienzo de banda en milímetros con arrastre, redimensionado, ajuste a cuadrícula y desplazamiento con el teclado; un inspector de propiedades generado a partir de `schema.json`; Guardar (con el campo problemático resaltado a partir de la ruta JSON de un error de validación del lado del servidor) y Vista previa; Deshacer y rehacer; y un botón Descargar JSON. Véase TsgcReportHTMLDesigner para la lista completa de propiedades, y TsgcReportHTTPEndpoint para las rutas del diseñador que llama (`schema.json`, `{name}/definition` y `{name}/preview.png`).

Editar una definición sin el diseñador del navegador

La demo `ReportJSONDefinition` (`delphi\Demos\Delphi\ReportJSONDefinition`) muestra el mismo ciclo completo desde código Delphi, útil para un editor de escritorio o para pruebas automatizadas de una definición: edita `definition.json` y su `data.json` en dos memos, un botón Render carga la definición con `sgcReportLoadFromJSON` y muestra la vista previa, los errores se muestran en la barra de estado junto con la ruta JSON del miembro problemático (véase Definiciones de informes JSON), y un botón Save definition escribe de nuevo el informe actual en JSON con `sgcReportSaveToJSON`.
El proyecto completo y compilable para el editor basado en código es `delphi\Demos\Delphi\ReportJSONDefinition\sgcReports_Demo_JSONDefinition.dpr`.

Introducción

sgcCharts es una biblioteca de componentes Delphi para gráficos de datos en vivo. Una fuente de datos añade puntos o velas OHLC a un búfer circular, y un lienzo dibuja una instantánea decimada de ese búfer, de modo que una fuente que publica miles de actualizaciones por segundo sigue renderizándose a una frecuencia de fotogramas estable.
La biblioteca se dirige a VCL, FireMonkey y renderizado sin interfaz visual (salida PNG o SVG sin ventana), así que el mismo código de búfer y de fuente de datos dibuja en un formulario, en una pantalla móvil o dentro de una aplicación de consola o un servicio.
Esta ayuda cubre las partes de sgcCharts que existen hoy: las fuentes de datos que alimentan un búfer (TsgcChartDataSource y sus descendientes), el puente sgcWebSockets que alimenta un búfer desde una conexión en vivo (Bridge overview), y los propios búferes y las funciones de decimación (TsgcChartBuffer).

- Información general: el modelo, los lienzos, los búferes y las fuentes de datos, y cómo encajan entre sí.
- Ediciones: qué incluyen Registered, Trial y Community.
- Inicio Rápido: estado actual del recorrido de introducción.
- Fuentes de Datos: manual, conjunto de datos, JSON y CSV.
- Puente: MQTT, WebSocket, SSE y Binance, a través de sgcWebSockets.
- Referencia: las clases de búfer y las funciones de decimación.

Arquitectura

sgcCharts separa los datos que muestra un gráfico de la superficie sobre la que se dibuja, de modo que los mismos datos pueden alimentar un formulario VCL, una vista FireMonkey o un render PNG/SVG sin interfaz visual.

Un modelo, varios lienzos

Los datos de un gráfico residen en un búfer, no en un control. **TsgcChartBuffer** guarda puntos X/Y y **TsgcChartOHLCBuffer** guarda velas OHLC (apertura, máximo, mínimo, cierre, volumen). Ambas son clases simples sin dependencia de VCL ni FireMonkey, así que pueden crearse, alimentarse y leerse desde una aplicación de consola, un servicio o un hilo en segundo plano. Dibujar es un paso independiente: un lienzo lee una instantánea decimada de un búfer y la pinta, usando un lienzo rasterizado (`sgcUICore_Canvas_Raster`) o un lienzo SVG (`sgcUICore_Canvas_SVG`) tras la superficie compartida `sgcUICore_Canvas`.
La capa de canvas, raster, fuente TrueType, SVG y temas que dibuja estos gráficos no forma parte de sgcCharts en sí. Es el paquete compartido **sgcUICore** (unidades Delphi `sgcUICore_*` en el paquete de ejecución sgcUICore, paquetes .NET `esegece.sgcUICore` y `esegece.sgcUICore.Rendering`). Se instala automáticamente con sgcCharts y se comparte con los demás productos visuales de eSeGeCe.

Búferes circulares con decimación min/max

Ambas clases de búfer son búferes circulares de capacidad fija. En modo `cbmOverwrite`, un búfer lleno descarta su punto más antiguo en la siguiente adición; en modo `cbmGrow`, su capacidad se duplica en su lugar. Todo método público de un búfer toma un bloqueo interno, así que un hilo de alimentación puede añadir mientras un hilo de render lee una instantánea sin que ambos colisionen. **Snapshot** y **SnapshotRange** copian los datos que necesita el llamador en un registro propiedad del llamador bajo ese bloqueo y lo liberan de inmediato; el llamador entonces decima y dibuja a partir de la copia, fuera del bloqueo.
`sgcChartDecimate` reduce una instantánea a aproximadamente un conjunto de puntos por columna de píxeles, usando `dmMinMax` (el primer, mínimo, máximo y último punto de cada columna, lo que mantiene una polilínea en vivo visualmente exacta) o `dmLTTB` (Largest Triangle Three Buckets, para una serie estática). `sgcChartDecimateOHLC` fusiona de la misma forma las velas que caen en la misma columna de píxeles. Véase TsgcChartBuffer y Decimación.

Gráfico, series y ejes

TsgcChart es el modelo de gráfico: un componente no visual que contiene los ejes, las series, el título, la leyenda, los márgenes, el tema y la paleta, pero no dibuja nada por sí mismo. Una serie es en sí misma un componente, no una parte anidada del gráfico; asignar su propiedad **Chart** es lo que la registra, y una serie lee su propio búfer directamente (a través de uno de los tipos de serie concretos) en lugar de pasar por el gráfico. Una vista llama a **TsgcChart.Render** para pintar todo el modelo en un TsgcUICanvas, de modo que el mismo gráfico, con las mismas series y ejes, puede mostrarse en más de una vista, o renderizarse sin interfaz visual a PNG o SVG sin ninguna vista en absoluto. Véase TsgcChart y TsgcChartSeries.

Fuentes de datos

Un descendiente de **TsgcChartDataSource** lee registros de algún origen (una aplicación que llama a **AppendValue** directamente, un `TDataSet`, un documento JSON o texto CSV) y añade los puntos o velas correspondientes a los búferes asignados a sus propiedades **Buffer** y **OHLCBuffer**. Dónde residen los campos de un registro se describe una sola vez, en **Mapping**, y toda fuente de datos lo lee de la misma forma. Véase Fuentes de Datos.

Puente sgcWebSockets

El paquete opcional `sgcChartsWS` añade fuentes de datos que leen desde una conexión sgcWebSockets en vivo: un tema MQTT, un flujo de mensajes WebSocket, un flujo de Server-Sent Events, o la API pública de datos de mercado de Binance. Cada una engancha el evento existente del componente cliente al que apunta, así que el maneja-

dor propio de la aplicación sigue disparándose junto con el gráfico. Este paquete requiere sgcWebSockets; el paquete básico `sgcCharts` no lo requiere. Véase Puente.

Ediciones

Véase Ediciones para saber qué incluyen Registered, Trial y Community.

Ediciones

sgcCharts se compila a partir del mismo código fuente para todas las ediciones; la edición solo cambia qué unidades incluye una compilación.

Edición	Incluye
Registered	Todas las unidades: las fuentes de datos básicas, los indicadores y paneles financieros, y las fuentes de datos en vivo (streaming), incluido el puente sgcWebSockets. Sin aviso de prueba.
Trial	Las mismas unidades que Registered. Una compilación hecha con `SGC_TRIAL` muestra un aviso único cuando ha pasado la ventana de la versión indicada en la compilación; nada más cambia.
Community	El núcleo completo de gráficos sin límites funcionales: las ocho series (línea, línea rápida, línea escalonada, área, barra/columna, dispersión, velas y OHLC), las vistas VCL, FMX y sin interfaz visual, los búferes, las fuentes de datos básicas (manual, conjunto de datos, JSON, CSV) e interacción. Una compilación hecha con `SGC_COMMUNITY` excluye el paquete financiero (paneles, series de indicadores, series de volumen, agregador, herramientas de dibujo, marcador de último valor y banda de oferta/demanda), y la edición Community no incluye las unidades ni los paquetes del puente sgcWebSockets.

La edición es una elección en tiempo de compilación en `sgcVer.inc` (`SGC_COMMUNITY`, que también desactiva `SGC_TRIAL`), no un interruptor en tiempo de ejecución.

Instalación

sgcCharts forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcCharts** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcCharts

- **sgcUICore**, la base compartida de sgcCharts y de los productos de disposición. No tiene casilla propia. El instalador la instala automáticamente siempre que se selecciona sgcCharts.
- **sgcCharts para sgcWebSockets**, que aparece en **sgcWebSockets bridges**. Instala los paquetes sgcChartsWS con los orígenes de datos en vivo del puente de sgcWebSockets. El instalador solo lo ofrece cuando sgcWebSockets ya está instalado en el equipo y su licencia incluye sgcCharts. De lo contrario aparece atenuado.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Charts`	Las demos de sgcCharts. Consulte Demostraciones.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluido sgcCharts.
- **Community**: el instalador gratuito Community de sgcUI incluye sgcCharts, junto con sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard. Los puentes de sgcWebSockets no forman parte del instalador Community, y sgcPDF y sgcMaps no forman parte de Community. Consulte Ediciones para saber qué deja fuera la compilación Community.

Migración desde una instalación independiente de sgcCharts

Si sgcCharts se instaló antes con su propio instalador independiente, el instalador de sgcUI detecta esa instalación, junto con la carpeta sgcUICore que escribió el instalador independiente. Muestra sus rutas de bibliotecas y sus paquetes de diseño en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta. La página solo se muestra cuando se encuentra una instalación antigua.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcChartsD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcCharts<V>` / `dclsgcCharts<V>`	Los componentes de sgcCharts (tiempo de ejecución / tiempo de diseño).
`sgcChartsWS<V>` / `dclsgcChartsWS<V>`	El puente de sgcWebSockets (tiempo de ejecución / tiempo de diseño). Requiere sgcWebSockets.

Compílelos en orden de dependencia: `sgcUICore`, después `sgcCharts`, después `dclsgcCharts` y, por último, los paquetes del puente. Después instale los paquetes de diseño `dcl*` y añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` correspondiente a su versión, por ejemplo `sgcChartsC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

.NET

No cambia nada para .NET. sgcCharts se sigue distribuyendo como sus propios paquetes NuGet (`esegece.sgcCharts` y sus paquetes complementarios). Consulte la información general de .NET.

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13. Los paquetes generan las cabeceras .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	TsgcVCLChart en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	TsgcFMXChart desde Delphi XE6 (SGC_CHARTS_FMX), en Win32, Win64, macOS, iOS y Android.
Linux64	Las unidades independientes del framework, TsgcChart y TsgcChartRenderer con la salida PNG y SVG sin interfaz, desde Delphi 10.2 (SGC_CHARTS_LINUX).
.NET	Un port en C# para WinForms, WPF y Avalonia, con salida PNG y SVG sin interfaz, véase Visión general de .NET.

Paquetes

sgcCharts incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcChartsD13.dpk para Delphi 13 y sgcChartsC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcCharts	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, dbrtl, sgcUICore
sgcChartsVCL	Runtime, VCL	sgcCharts_VCL (TsgcVCLChart)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcCharts
sgcChartsFMX	Runtime, FireMonkey, desde XE6	sgcCharts_FMX (TsgcFMXChart)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcCharts
dclsgcCharts	Tiempo de diseño, VCL	Registra TsgcChart, TsgcChartRenderer y TsgcVCLChart en la página de paleta **SGC Charts**.	designide, rtl, vcl, sgcCharts, sgcChartsVCL
dclsgcChartsFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXChart en la página de paleta **SGC Charts**.	designide, rtl, fmx, sgcCharts, sgcChartsFMX, dclsgcCharts
sgcChartsWS, dclsgcChartsWS	Runtime y tiempo de diseño, puente de sgcWebSockets	El puente de datos en vivo, independiente del framework.	rtl, sgcCharts, sgcWebSockets

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcChartsFMX requiere dclsgcCharts, así que instale primero dclsgcCharts.

Versiones del IDE y plataformas de destino

Delphi	sgcCharts	sgcCharTsVCL	sgcChartsFMX	dclsgcCharts, dclsgcChartsFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcCharts, sgcCharTsVCL	sgcChartsFMX	dclsgcCharts, dclsgcCharTsFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcCharts)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcCharts)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcChartsC13.cbproj, sgcChartsVCLC13.cbproj, sgcChartsFMXC13.cbproj, dclsgcChartsC13.cbproj y dclsgcChartsFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXChart) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcChartsVCL solo funcionan en Windows.

- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcCharts_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcCharts se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa TsgcChart y TsgcChartRenderer, que renderiza un gráfico en PNG o SVG sin ventana. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Véase también

- Instalación
- Ediciones
- Arquitectura
- Solución de problemas

Inicio Rápido

sgcCharts separa el **modelo** del gráfico de la **vista** que lo dibuja. Un TsgcChart contiene los ejes, las series y los datos; no dibuja nada por sí mismo. Una vista, un control VCL, un control FireMonkey o un renderizador sin interfaz visual, pinta ese modelo cada vez que cambia. Este recorrido construye un gráfico paso a paso: primero en un formulario VCL, después alimentado con datos en vivo, y por último renderizado sin ninguna vista en absoluto para un servicio o una aplicación de consola.

1. El modelo de gráfico y la vista VCL

Cree un **TsgcChart** y dele un eje inferior y uno izquierdo, después cree un **TsgcVCLChart** y apunte su propiedad **Chart** al modelo. La vista se registra a sí misma como oyente y se repinta cada vez que el modelo cambia; no se necesita nada más para ver un gráfico (vacío) en el formulario.

```
uses
  Forms, Classes, SysUtils, Controls,
  sgcCharts_Chart, sgcCharts_Types,
  sgcCharts_VCL;

type
  TfrmMain = class(TForm)
  private
    FChart: TsgcChart;
    FView: TsgcVCLChart;
  public
    constructor Create(aOwner: TComponent); override;
  end;

constructor TfrmMain.Create(aOwner: TComponent);
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcCharts Quick Start';
  Width := 820;
  Height := 480;

  FChart := TsgcChart.Create(Self);
  FChart.Axes.Add.Name := 'bottom';
  FChart.Axes[0].Position := capBottom;
  FChart.Axes.Add.Name := 'left';
  FChart.Axes[1].Position := capLeft;

  FView := TsgcVCLChart.Create(Self);
  FView.Parent := Self;
  FView.Align := alClient;
  FView.Chart := FChart;
end;
```

2. Añadir una serie y datos estáticos

Una serie es un componente por derecho propio. Cree un TsgcChartSeries_Line, apunte su propiedad **Chart** al modelo, nombre los ejes contra los que se representa, y llame a **AddXY** una vez por punto. Añada las dos líneas siguientes al constructor de arriba, después de crear la vista.

```
uses
  // ... as above, plus:
  Math,
  sgcCharts_Series;

// inside TfrmMain, add a field:
//   FSeries: TsgcChartSeries_Line;
```

```
// inside the constructor, after FView.Chart := FChart:
var
  i: Integer;
begin
  FSeries := TsgcChartSeries_Line.Create(Self);
  FSeries.Chart := FChart;
  FSeries.AxisX := 'bottom';
  FSeries.AxisY := 'left';
  for i := 0 to 99 do
    FSeries.AddXY(i, Sin(i / 10) * 50 + 50);
end;
```

3. Alimentar datos en vivo

Para seguir añadiendo puntos después de mostrar el formulario, coloque un `TTimer` y llame a **AddY** desde su evento **OnTimer**. **AddY** añade un punto en el siguiente valor X consecutivo, lo que resulta conveniente cuando X es un índice de muestra o un recuento de ciclos en lugar de un valor que el llamador ya tiene.

```
uses
  // ... as above, plus:
  ExtCtrls;

// inside TfrmMain, add fields:
//   FTimer: TTimer;
//   FTicks: Integer;
// and a method:
//   procedure DoTimer(aSender: TObject);

// inside the constructor, after the series loop:
FTimer := TTimer.Create(Self);
FTimer.Interval := 50;
FTimer.OnTimer := DoTimer;
FTimer.Enabled := True;

procedure TfrmMain.DoTimer(aSender: TObject);
begin
  Inc(FTicks);
  FSeries.AddY(Sin(FTicks / 5) * 50 + 50);
end;
```

Un temporizador que llama a **AddY** sirve para una prueba rápida, pero una aplicación real que represente una fuente en vivo debería usar en su lugar uno de los componentes fuente de datos: **TsgcChartDataSource_Manual** para valores que la aplicación ya tiene en la mano, **TsgcChartDataSource_DataSet** para un `TDataSet`, o **TsgcChartDataSource_JSON** / **TsgcChartDataSource_CSV** para un documento o un archivo. Añaden adiciones seguras para hilos, notificaciones con limitación de frecuencia y (para la fuente de conjunto de datos) asignación de campos que un temporizador simple no ofrece. Para una fuente que ya llega por MQTT, un WebSocket, Server-Sent Events o un flujo de mercado de Binance, las fuentes de datos del puente se conectan directamente al cliente sgcWebSockets correspondiente y no necesitan ningún sondeo en absoluto.
Juntos, los pasos 1 a 3 dan el programa completo y compilable de abajo. El formulario no usa ningún `.dfm`, construye su propio diseño en código, por lo que el constructor llama a **CreateNew** en lugar del habitual **Create** heredado.

```
program QuickStartVCL;

{$APPTYPE GUI}

uses
  Forms, Classes, SysUtils, Math, ExtCtrls, Controls,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcCharts_VCL;

type
  TfrmMain = class(TForm)
  private
```

```
    FChart: TsgcChart;
    FView: TsgcVCLChart;
    FSeries: TsgcChartSeries_Line;
    FTimer: TTimer;
    FTicks: Integer;
    procedure DoTimer(aSender: TObject);
  public
    constructor Create(aOwner: TComponent); override;
  end;

constructor TfrmMain.Create(aOwner: TComponent);
var
  i: Integer;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcCharts Quick Start';
  Width := 820;
  Height := 480;

  FChart := TsgcChart.Create(Self);
  FChart.Axes.Add.Name := 'bottom';
  FChart.Axes[0].Position := capBottom;
  FChart.Axes.Add.Name := 'left';
  FChart.Axes[1].Position := capLeft;

  FView := TsgcVCLChart.Create(Self);
  FView.Parent := Self;
  FView.Align := alClient;
  FView.Chart := FChart;

  FSeries := TsgcChartSeries_Line.Create(Self);
  FSeries.Chart := FChart;
  FSeries.AxisX := 'bottom';
  FSeries.AxisY := 'left';
  for i := 0 to 99 do
    FSeries.AddXY(i, Sin(i / 10) * 50 + 50);

  FTicks := 0;
  FTimer := TTimer.Create(Self);
  FTimer.Interval := 50;
  FTimer.OnTimer := DoTimer;
  FTimer.Enabled := True;
end;

procedure TfrmMain.DoTimer(aSender: TObject);
begin
  Inc(FTicks);
  FSeries.AddY(Sin(FTicks / 5) * 50 + 50);
end;

var
  vForm: TfrmMain;
begin
  Application.Initialize;
  Application.CreateForm(TfrmMain, vForm);
  Application.Run;
end.
```

4. Renderizado sin interfaz visual, sin ninguna vista

Un servicio, un módulo ISAPI o una aplicación de consola no tiene ningún formulario en el que colocar una vista, y no lo necesita: el mismo **TsgcChart** y sus series se renderizan a través de un TsgcChartRenderer en lugar de un **TsgcVCLChart**. **TsgcChartRenderer** no tiene dependencia de VCL ni FMX, así que este programa no se enlaza con nada más que las unidades básicas de sgcCharts.

```
program QuickStartHeadless;

{$APPTYPE CONSOLE}

uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcCharts_Renderer;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  vRenderer: TsgcChartRenderer;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Title.Text := 'Quick Start (headless)';
    vChart.Axes.Add.Name := 'bottom';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'left';
    vChart.Axes[1].Position := capLeft;

    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Chart := vChart;
      vSeries.AxisX := 'bottom';
      vSeries.AxisY := 'left';
      for i := 0 to 99 do
        vSeries.AddXY(i, Sin(i / 10) * 50 + 50);

      vRenderer := TsgcChartRenderer.Create(nil);
      try
        vRenderer.Chart := vChart;
        vRenderer.Width := 800;
        vRenderer.Height := 400;
        vRenderer.RenderToPNGFile('quickstart.png');
      finally
        vRenderer.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Al ejecutarlo se escribe `quickstart.png` junto al ejecutable, sin ventana, sin bucle de mensajes y sin necesitar escritorio. Véase TsgcChartRenderer para renderizar a SVG, a un flujo, o a un búfer de píxeles en bruto en su lugar.

Hacia dónde ir después

- Series: línea, línea rápida, línea escalonada, área, barra, dispersión y velas, y qué dibuja cada una.
- Vistas: **TsgcVCLChart**, **TsgcFMXChart** y **TsgcChartRenderer**.
- Interacción: zoom, desplazamiento, retícula e información sobre herramientas en una vista visual.
- Financiero: indicadores, osciladores, volumen y gráficos financieros multipanel.
- Fuentes de Datos: fuentes manuales, de conjunto de datos, JSON y CSV.
- Fuentes de Datos del Puente: fuentes en vivo de MQTT, WebSocket, SSE y Binance a través de sgcWebSockets.
- Demostraciones: los proyectos de demostración ejecutables en los que se basa este inicio rápido.

Véase también

- TsgcChart

- TsgcChartSeries
- TsgcVCLChart
- TsgcChartRenderer
- TsgcChartBuffer

Demostraciones

La distribución de sgcCharts incluye un proyecto de demostración para cada escenario de esta ayuda: una fuente en vivo, varios ejes, un gráfico de velas a partir de un CSV, un gráfico vinculado a un conjunto de datos, temas e HiDPI, una fuente en vivo conectada desde sgcWebSockets, un gráfico móvil FireMonkey, y renderizado sin interfaz visual sin ninguna vista en absoluto. Los proyectos Delphi residen en `Demos\Delphi\<Nombre>\` y sus contrapartidas de C++Builder en `Demos\CBuilder\<Nombre>\`; toda carpeta de demostración incluye un `readme.html` con los detalles.

Demostraciones VCL

Estas demostraciones colocan un **TsgcChart**, un **TsgcVCLChart**, las series y las fuentes de datos en un formulario en tiempo de diseño. Se compilan con Delphi 7 a Delphi 13, Win32 y Win64.

Demostración	Qué muestra	Carpeta
LiveLine	Un **TsgcChartSeries_FastLine** con un búfer circular de 1.000.000 de puntos, alimentado desde un hilo en segundo plano a través de un **TsgcChartDataSource_Manual** entre 1.000 y 200.000 puntos por segundo, con un **TimeWindow** de 10 s / 60 s / búfer completo, un selector de tema claro/oscuro, y una retícula e información sobre herramientas sobre un millón de puntos.	Delphi\LiveLine\
MultiAxis	Tres series de línea en tres ejes de valor (izquierda, derecha y un segundo eje izquierdo), un eje inferior de fecha y hora y la leyenda a la derecha; cada serie elige sus ejes con **AxisX** y **AxisY**, con zoom de rueda, zoom de caja, desplazamiento y reinicio en cada eje.	Delphi\MultiAxis\
Candlestick	Un año de datos OHLCV diarios de un CSV compilado en el ejecutable, cargado por **TsgcChartDataSource_CSV** en una serie de velas con el volumen como barras en un segundo eje; los nombres de columna se asignan a X, apertura, máximo, mínimo, cierre y volumen con fechas ISO 8601, y un interruptor alterna entre velas o barras OHLC.	Delphi\Candlestick\
DataSetChart	Un `TClientDataSet` con 1.000 filas, mostrado en un `TDBGrid` y leído por dos componentes **TsgcChartDataSource_DataSet** en una serie de línea y una serie de barras; **XField** y **YField** nombran los campos del conjunto de datos, y editar una fila en la cuadrícula y recargar actualiza el gráfico.	Delphi\DataSetChart\
Themes	El mismo gráfico en el tema claro y en el oscuro lado a lado, con un selector de escala (1.0, 1.5, 2.0) y exportación a PNG y SVG a través de **TsgcVCLChart.SaveToPNG** y **SaveToSVG**. El proyecto es `Themes\ChartThemes.dpr` .	Delphi\Themes\
PieDonut	Un **TsgcChartSeries_Pie** y un anillo lado a lado, alimentados en vivo desde un pequeño `IsgcUITable` editado en una cuadrícula de cadenas mediante **TsgcChartDataSource_Table**, con un control deslizante del umbral Otros, explotar al hacer clic, un selector del modo de etiquetas y exportación a PNG y SVG.	Delphi\PieDonut\
RadarCompare	Tres productos en ocho criterios con **TsgcChartSeries_Radar**: relleno o contorno, cuadrícula poligonal o circular, y un temporizador que mueve las puntuaciones.	Delphi\RadarCompare\

HeatmapLive	Un espectrograma **TsgcChartSeries_Heatmap** que añade 30 líneas por segundo sobre un eje de fecha y hora con **Capacity** 300, y una matriz de región por mes cargada desde un `IsgcUITable` con ejes de categorías, etiquetas de valor y una escala de color continua o escalonada.	Delphi\HeatmapLive\
TreemapFunnel	Un **TsgcChartSeries_Treemap** de tres niveles con navegación en profundidad y ruta de navegación, y un **TsgcChartSeries_Funnel** de ventas en vivo de seis etapas.	Delphi\TreemapFunnel\

Demostraciones del Puente sgcWebSockets

Estas demostraciones alimentan los gráficos con datos en vivo desde componentes sgcWebSockets a través de las fuentes de datos del puente. Necesitan tener instalado sgcWebSockets y se compilan con Delphi 2010 a Delphi 13, Win32 y Win64. Los clientes y servidores sgcWebSockets y las fuentes de datos se colocan en el formulario en tiempo de diseño.

Demostración	Qué muestra	Carpeta
MQTTSensors	Tres **TsgcChartDataSource_MQTT** comparten un `TsgcWSPClient_MQTT` y representan los temas `sgccharts/sensors/temperature`, `humidity` y `pressure`, suscritos una sola vez a `sgccharts/sensors/+`, contra test.mosquitto.org por WebSocket o un broker local por TCP simple; un botón Simulate publica valores aleatorios cada 100 ms para que la demostración funcione contra cualquier broker, y la presión obtiene su propio eje derecho.	Delphi\MQTTSensors\
BinanceLive	Dos **TsgcChartDataSource_Binance** sobre un `TsgcWSAPI_Binance`: un flujo de klines alimenta una serie Candlestick que actualiza la vela en formación con cada mensaje, y un flujo de operaciones alimenta una serie Line; las últimas 60 velas se cargan a través de la API REST de Binance antes de suscribirse, y el símbolo (BTCUSDT o ETHUSDT) y el intervalo (1m o 5m) pueden cambiarse en caliente.	Delphi\BinanceLive\
WebSocketFeed	Un `TsgcWebSocketServer` dentro de la demostración difunde `{"t":<unix ms>,"v":<value>}` cada 20 ms, y un **TsgcChartDataSource_WebSocket** sobre un `TsgcWebSocketClient` lo representa, leyendo **XPath** `t` como milisegundos Unix e **YPath** `v`; el servidor y el cliente se ejecutan en el mismo programa de bucle local en 127.0.0.1.	Delphi\WebSocketFeed\
SSEFeed	Un flujo de eventos servido dentro de la demostración, leído con `TsgcHTTP1Client.GetSSE` en un hilo de trabajo y representado por un **TsgcChartDataSource_SSE**, que divide el flujo en bruto en eventos y lee sus datos JSON mientras el propio **OnSSEMessage** del formulario sigue disparándose y cancela el flujo al detenerse.	Delphi\SSEFeed\

Demostraciones FMX

Esta demostración coloca un TsgcFMXChart en un formulario FireMonkey. Se compila con Delphi 12 y 13, para Win32, Win64, macOS (OSX64, OSXARM64), iOSDevice64 y Android64.

Demos-	Qué muestra	Carpeta

tra-ción		
FMX-Mobile	Un control de pestañas con un **TsgcChartSeries_FastLine** en vivo (un búfer circular de 200.000 puntos alimentado desde un hilo en segundo plano a 20.000 puntos por segundo) y un **TsgcChartSeries_Candlestick** con 500 velas, dimensionado para tacto; zoom de pellizco y desplazamiento en el eje de tiempo, la retícula y la información sobre herramientas, un botón Reset, un **TimeWindow** de 10 s / 30 s, un selector de tema claro/oscuro, y fotogramas por segundo desde **TsgcFMXChart.FrameCount**.	Delphi\FMXMobile\

Demostraciones de Renderizado sin Interfaz Visual

Estas demostraciones renderizan gráficos sin ningún control visual, a través de TsgcChartRenderer, que no tiene dependencia de VCL ni FMX y por tanto no necesita tener ninguno instalado. Se ejecutan en programas de consola, servicios de Windows, módulos ISAPI y daemons Linux64.

De-mos-tra-ción	Qué muestra	Carpeta
Head-less	Un programa de consola que construye en memoria una serie Line de 10.000 puntos y una serie Candlestick de 500 velas, y renderiza cada una a un archivo PNG y SVG (`line.png` , `line.svg` , `candles.png` , `candles.svg`), imprimiendo el tamaño y el CRC32 de cada archivo. Los datos se generan mediante aritmética de enteros y el rasterizador es idéntico bit a bit entre plataformas, así que los archivos PNG escritos en Win64 y en Linux64 comparten el mismo CRC32. Se ejecuta en Win32, Win64 y Linux64.	Delphi\Headless\
Head-less-Servi-ce	Un servicio de Windows, instalado con `HeadlessService.exe /install` y desinstalado con `/uninstall` , que añade un valor a un gráfico de línea y lo renderiza en `output\chart.png` cada 10 segundos; como un servicio no tiene escritorio, el gráfico lo dibuja **TsgcChartRenderer** en lugar de un control visual, y los errores de render van al registro de eventos de Windows. Se ejecuta en Win32 y Win64.	Delphi\HeadlessService\

Véase también

- Inicio Rápido
- Fuentes de Datos
- Fuentes de Datos del Puente
- TsgcChartRenderer

TsgcChart

Unidad: sgcCharts_Chart
TsgcChart es el modelo de gráfico: un descendiente de `TComponent` que contiene los ejes, las series, el título, subtítulo, leyenda y márgenes y el tema y paleta, pero no dibuja nada por sí mismo y no depende de ningún control visual. Una vista (un control VCL o FireMonkey, o un renderizador sin interfaz) llama a **AddListener** para saber cuándo ha cambiado el modelo y a **Render** para pintarlo en un TsgcUICanvas. Como el modelo y el dibujo están separados, el mismo TsgcChart puede mostrarse en más de una vista a la vez, o representarse directamente en un archivo PNG o SVG sin ninguna vista.

Propiedades

Propiedad	Tipo	Descripción
Theme	TsgcUITheme	`utLight`, `utDark` o `utCustom`. Predeterminado `utLight`. Véase Temas y Paleta.
Axes	TsgcChartAxes	La colección de ejes. Véase Ejes.
Title, Subtitle	TsgcChartTitle	Véase Título, Leyenda y Márgenes.
Legend	TsgcChartLegend	Véase Título, Leyenda y Márgenes.
Margins	TsgcChartMargins	Véase Título, Leyenda y Márgenes.
BackgroundColor, PlotBackgroundColor	TsgcUIColor	Se almacena solo cuando **Theme** es `utCustom`. Establecer cualquiera de las dos directamente cambia **Theme** a `utCustom`, salvo que el componente se esté cargando desde un flujo.
Palette	TsgcChartPalette	Los 8 colores automáticos de las series. Véase Temas y Paleta.
MaxFPS	Integer	Límite de repintado que se espera que respeten las vistas. Predeterminado 60.
TimeWindow	Double	Segundos, predeterminado 0 (desactivado). Mientras no sea cero, un eje inferior automático numérico o de fecha y hora muestra `[newest X - TimeWindow, newest X]` en lugar del rango de datos completo.
PlotRect	TsgcUIRect (read-only)	El rectángulo de trazado calculado por el último **Render**.
Scale	Double	Factor HiDPI. La vista lo establece antes de llamar a **Render**; cambiarlo no genera ninguna notificación.
SeriesCount	Integer (read-only)	Número de series cuya propiedad **Chart** apunta a este gráfico.
Series[Index: Integer]	TsgcChartSeries (read-only, default)	La serie en Index, basada en 0.

Métodos

Método	Devuelve	Descripción
BeginUpdate / EndUpdate		Agrupa varios cambios para que los agentes de escucha se notifiquen una sola vez, en **EndUpdate**, en lugar de después de cada cambio.
Invalidate		Notifica ahora a los agentes de escucha, o difiere la notificación hasta el **EndUpdate** correspondiente cuando hay una actualización abierta.
AddListener(aEvent: TNotifyEvent)		Registra un evento a través del cual una vista recibe la llamada cuando el gráfico necesita repintarse.
RemoveListener(aEvent: TNotifyEvent)		Anula el registro de un agente de escucha añadido con **AddListener**.
Render(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect)		Dibuja el fondo, el título, el subtítulo, la leyenda, los ejes y cada serie visible dentro de **aRect** (píxeles de dispositivo) de **aCanvas**. Véase Canvas.
FindAxis(const aName: string)	TsgcChartAxis	El eje con este Name (sin distinguir mayúsculas y minúsculas), nil cuando no hay ninguno.
AxisRect(aAxis: TsgcChartAxis)	TsgcUIRect	La banda que el último **Render** reservó para aAxis fuera del trazado; un rectángulo vacío cuando se desconoce.
AreaCount, AreaRect(aIndex), AreaMode(aIndex), AreaAt(X, Y)	Integer, TsgcUIRect, TsgcChartPlotMode, Integer	Las áreas de trazado del último **Render**, sus rectángulos y modos, y el área bajo un píxel. Véase Áreas de trazado.
SeriesAreaRect(aSeries), IsSeriesDrawn(aSeries)	TsgcUIRect, Boolean	Dónde dibujó el último **Render** una serie, y si llegó a dibujarla.

Modelo y vistas

TsgcChart no lleva ninguna superficie de dibujo propia: es un componente sencillo, no un control, por lo que puede construirse, poblarse y renderizarse desde un servicio o una aplicación de consola exactamente igual que desde un formulario. Las series son a su vez componentes con una propiedad **Chart** (véase TsgcChartSeries); asignar esa propiedad registra la serie en el gráfico, y vaciarla (o destruir la serie) la elimina. Una vista solo necesita tres cosas del modelo: **AddListener** para saber cuándo repintar, **Render** para realizar el dibujo, y **Scale** establecido con su factor HiDPI de antemano.

Actualizaciones y notificaciones

Cada cambio en el gráfico, sus series o sus ejes llama a **Invalidate**, que notifica de inmediato a los agentes de escucha registrados. Envolver varios cambios entre **BeginUpdate** y **EndUpdate** difiere esa notificación hasta el **EndUpdate** correspondiente, de modo que una vista repinta una sola vez en lugar de una vez por cada cambio.

Ejemplo: Construcción y renderizado de un gráfico en código

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster, sgcUICore_Canvas_SVG;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  vAxisX, vAxisY: TsgcChartAxis;
  vRaster: TsgcUICanvas_Raster;
  vSVG: TsgcUICanvas_SVG;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Title.Text := 'Monthly Revenue';

    vAxisX := vChart.Axes.Add;
    vAxisX.Name := 'x';
    vAxisX.Position := capBottom;

    vAxisY := vChart.Axes.Add;
    vAxisY.Name := 'y';
    vAxisY.Position := capLeft;

    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Chart := vChart;
      vSeries.AxisX := 'x';
      vSeries.AxisY := 'y';
      for i := 0 to 99 do
        vSeries.AddXY(i, Sin(i / 10) * 100);

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Scale := 1;
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
        vRaster.SaveToPNGFile('chart.png');
      finally
        vRaster.Free;
      end;

      vSVG := TsgcUICanvas_SVG.Create(800, 400);
      try
        vChart.Render(vSVG, sgcUIRect(0, 0, 800, 400));
        vSVG.SaveToFile('chart.svg');
      finally
        vSVG.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChartSeries
- Ejes

- Áreas de trazado
- Canvas

TsgcChartSeries

Unidad: sgcCharts_Chart
TsgcChartSeries es la base abstracta de cada serie: un descendiente de `TComponent` que un TsgcChart muestra mediante su propiedad **Chart**. Define el color, la visibilidad, la leyenda y las propiedades de enlace a ejes que comparten todas las series, y dos miembros abstractos, **GetDataRange** y **Draw**, que un tipo concreto implementa. Los tipos de serie concretos (línea, área, línea escalonada, barra, dispersión, velas, OHLC y el resto) son componentes por derecho propio, declarados en `sgcCharts_Series` y documentados por separado.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El gráfico que muestra esta serie. Asignarlo registra la serie en el gráfico; vaciarlo, o establecerlo en un gráfico diferente, la anula del registro del anterior primero.
Title	string	Texto mostrado en la leyenda. Véase **LegendText**.
Color	TsgcUIColor	0 (el valor predeterminado) significa un color de paleta automático. Véase **EffectiveColor**.
Visible	Boolean	Predeterminado True.
ShowInLegend	Boolean	Predeterminado True.
AxisX, AxisY	string	El **Name** del eje contra el que se dibuja esta serie. Véase Ejes.
Decimation	TsgcChartDecimation	`dmNone` , `dmMinMax` o `dmLTTB` . Predeterminado `dmMinMax` . Véase Decimación.
Index	Integer (read-only)	Posición de la serie en la lista de series del gráfico, -1 cuando no está enlazada a un gráfico.

Métodos

Método	Devuelve	Descripción
Changed		Pide a **Chart** que se repinte (llama a su **Invalidate**); no hace nada cuando **Chart** es nil.
EffectiveColor	TsgcUIColor	**Color**, o el color de la paleta del gráfico en **Index** cuando **Color** es 0.
XAxis	TsgcChartAxis	El eje indicado por **AxisX**, resuelto en **Chart**; el primer eje inferior cuando **AxisX** está vacío, nil cuando no se encuentra.

YAxis	TsgcChartAxis	El eje indicado por **AxisY**, resuelto en **Chart**; el primer eje izquierdo cuando **AxisY** está vacío, nil cuando no se encuentra.
GetDataRange(var aXMin, aXMax, aYMin, aYMax: Double)	Boolean (abstract)	Los extremos de datos de la serie; False cuando no tiene datos. Implementado por el tipo de serie concreto.
Draw(aCanvas: TsgcUICanvas; aXAxis, aYAxis: TsgcChartAxis; const aPlot: TsgcUIRect; aScale: Double)		Abstracto. Dibuja la serie dentro de **aPlot**, ya recortada a él; los valores se asignan con **aXAxis.ValueToPixel** y **aYAxis.ValueToPixel**.
DrawLegendSymbol(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Dibuja el símbolo de la leyenda. El valor predeterminado es un cuadrado relleno de **EffectiveColor**; los tipos concretos lo sobrescriben.
LegendText	string	**Title** cuando está establecido, o si no, el **Name** del componente.
PlotMode	TsgcChartPlotMode (virtual)	Modo que esta serie da a su área de trazado: `cpmXY` (predeterminado), `cpmPolar` o `cpmCell`. Véase Áreas de trazado.
DrawArea(aCanvas; const aRect; const aSeries: TsgcChartSeriesArray; aCount: Integer; aScale: Double)	(virtual)	Pinta un área polar (se llama en el primer radar del área con todos los radares dibujados en **aSeries**) o una celda (**aSeries** contiene solo esta serie). El lienzo se recorta al área. La implementación predeterminada no hace nada.
HitTestArea(const aRect; const aSeries; aCount: Integer; X, Y: Double; var aHit: TsgcChartHitInfo)	Boolean (virtual)	Prueba de impacto de un área polar o de una celda, con los argumentos de **DrawArea**. La implementación predeterminada devuelve False.
HitTestPoint(const aPlot: TsgcUIRect; X, Y: Double; var aHit: TsgcChartHitInfo)	Boolean (virtual)	Prueba de impacto de una serie XY que la interacción no conoce, como el mapa de calor. La implementación predeterminada devuelve False.
TooltipText(const aHit: TsgcChartHitInfo)	string (virtual)	Información sobre herramientas de un impacto de esta serie; una cadena vacía (el valor predeterminado) usa el formato de la interacción. Varias líneas se dibujan como líneas simples.
AreaClick(const aHit: TsgcChartHitInfo)	(virtual)	Un clic en un impacto de esta serie, después de **OnSeriesClick**. Aquí se ejecutan explotar (gráfico circular) y profundizar (treemap). La implementación predeterminada no hace nada.
LegendItemCount	Integer (virtual)	Entradas de leyenda de la serie, 1 de forma predeterminada; un gráfico circular devuelve una por sector.
MeasureLegendItem(aCanvas; aItem: Integer; const aFont; aScale: Double; out aWidth, aHeight: Double)	(virtual)	Tamaño de la entrada de leyenda **aItem**. El valor predeterminado es el símbolo, un espacio y **LegendText** en una línea.
DrawLegendItem(aCanvas; aItem: Integer; const aRect; const aFont; aScale: Double)	(virtual)	Pinta la entrada de leyenda **aItem** dentro de **aRect**. El valor predeterminado llama a **DrawLegendSymbol** y dibuja **LegendText**.
GetDataCategories(aAxis: TsgcChartAxis; aList: TStrings)	Boolean (virtual)	Textos de categoría en tiempo de ejecución para un eje de categorías sin **Categories**; True cuando se

		rellenó **aList**. El mapa de calor lo usa para las etiquetas de sus líneas y celdas. La implementación predeterminada devuelve False.

Ganchos del área de trazado

Todos los ganchos usan de forma predeterminada el comportamiento XY, de modo que una serie que solo sustituye **GetDataRange** y **Draw** sigue siendo una serie XY. Las series de gráfico circular, radar, embudo y treemap sustituyen **PlotMode**, **DrawArea** y **HitTestArea**; el mapa de calor sigue siendo una serie XY y sustituye **HitTestPoint** y **GetDataCategories**. Véase Áreas de trazado para saber cómo se elige un modo.

Enlace al gráfico y a los ejes

Una serie no es propiedad de su gráfico: establecer **Chart** solo la registra, de modo que la aplicación sigue siendo propietaria y encargada de liberar la instancia de la serie (o de darle un propietario a la manera habitual de Delphi). **AxisX** y **AxisY** enlazan la serie a uno de los ejes de **Chart.Axes** por **Name** (sin distinguir mayúsculas y minúsculas); un nombre vacío recurre al primer eje inferior para X y al primer eje izquierdo para Y, de modo que un gráfico con un eje de cada tipo no necesita que todas las series los nombren.

Color y la paleta

Dejar **Color** en 0 hace que la serie tome el color de **Palette** del gráfico que corresponde a su **Index**, recorriendo cíclicamente los 8 colores de la paleta. Establecer **Color** en un valor distinto de cero sobrescribe la paleta para esa serie concreta. Véase Temas y Paleta.

Decimation

Decimation controla cómo una serie concreta reduce un búfer grande hasta aproximadamente un conjunto de puntos por columna de píxeles antes de dibujar. Véase Decimación para saber qué conserva cada modo.

Ejemplo: Una serie de línea enlazada a ejes con nombre

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  vBottom, vLeft: TsgcChartAxis;
  i: Integer;
  vRaster: TsgcUICanvas_Raster;
begin
  vChart := TsgcChart.Create(nil);
  try
    // ... keep the axes Axes.Add returns: Axes[0] and Axes[1] are the
    // default X and Y axes, not the new ones; no series uses the defaults
    // here, so they are hidden
    vChart.Axes.Find('X').Visible := False;
    vChart.Axes.Find('Y').Visible := False;
    vBottom := vChart.Axes.Add;
    vBottom.Name := 'bottom';
    vBottom.Position := capBottom;
    vLeft := vChart.Axes.Add;
    vLeft.Name := 'left';
    vLeft.Position := capLeft;

    vSeries := TsgcChartSeries_Line.Create(nil);
```

```
    try
      vSeries.Chart := vChart;
      vSeries.Title := 'Signal';
      vSeries.AxisX := 'bottom';
      vSeries.AxisY := 'left';
      for i := 0 to 199 do
        vSeries.AddXY(i, Sin(i / 15) * 50 + 50);

      Writeln('series index: ', vSeries.Index);
      Writeln('effective colour: $', IntToHex(vSeries.EffectiveColor, 8));

      vRaster := TsgcUICanvas_Raster.Create(600, 300);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 600, 300));
        vRaster.SaveToPNGFile('series.png');
      finally
        vRaster.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChart
- Ejes
- Decimación
- Áreas de trazado

TsgcChartAxis y TsgcChartAxes

Unidad: sgcCharts_Axis
TsgcChartAxes es la `TOwnedCollection` expuesta como TsgcChart.**Axes**; cada elemento es un **TsgcChartAxis** (un `TCollectionItem`). Un gráfico puede contener más de un eje en el mismo lado, que es como un gráfico muestra dos escalas de valores a la vez: las series se enlazan cada una a un eje por su nombre (véase TsgcChartSeries).

Propiedades de TsgcChartAxis

Propiedad	Tipo	Descripción
Name	string	Identificador al que se enlazan las series mediante **AxisX** / **AxisY**, y que busca **TsgcChart.FindAxis**.
AxisType	TsgcChartAxisType	`catNumeric` (predeterminado), `catLogarithmic`, `catDateTime` o `catCategory`.
Position	TsgcChartAxisPosition	`capLeft` (predeterminado), `capRight`, `capTop` o `capBottom`. Los ejes superior e inferior son horizontales (X); los ejes izquierdo y derecho son verticales (Y). Véase **IsHorizontal**.
Title, TitleStyle	string, TsgcUITextStyle	Título del eje y su estilo de texto.
LabelStyle	TsgcUITextStyle	Estilo de texto de las etiquetas de las marcas.
Visible	Boolean	Predeterminado True.
Automatic	Boolean	Predeterminado True. Cuando es True, el rango procede de los datos de las series enlazadas a este eje. Cuando es False, se utilizan **Minimum** / **Maximum** tal como se hayan indicado.
Minimum, Maximum	Double	Rango manual, utilizado cuando **Automatic** es False.
Inverted	Boolean	Predeterminado False.
LabelFormat	string	Vacío para el formato automático. Para `catNumeric` / `catLogarithmic`, una máscara de `FormatFloat` independiente de la configuración regional (`'0.00'`, `'#,##0.0'`). Para `catDateTime`, las letras de formato de fecha y hora indicadas más abajo.
GridVisible, GridColor, LineColor	Boolean, TsgcUIColor	Visibilidad de las líneas de cuadrícula y los colores de la cuadrícula y de la línea del eje. **GridVisible** predeterminado True.
TickLength	Double	Longitud de la marca de graduación en píxeles de dispositivo a escala 1.
MinTickSpacing	Double	Mínimo de píxeles de dispositivo a escala 1 entre dos marcas. Predeterminado 50.
Categories	TStrings	Etiquetas de marca para un eje `catCategory`, una por posición.

Métodos de TsgcChartAxis

Método	Devuelve	Descripción
IsHorizontal	Boolean	True cuando **Position** es `capTop` o `capBottom`.
SetDataRange(aMin, aMax: Double; aExact: Boolean = False)		Comunica los extremos de datos a partir de los cuales un eje automático debe ampliarse hasta valores redondos; **aExact** muestra **[aMin, aMax]** tal como se indica, sin ampliación (utilizado para **TimeWindow**).
ClearDataRange		Borra los extremos establecidos por **SetDataRange**.
SetDataCategories(aList: TStrings)		Textos de categoría en tiempo de ejecución de un eje de categorías sin **Categories**; el gráfico los establece durante el renderizado a partir de **GetDataCategories** de las series, como **SetDataRange**. Nunca se transmiten al flujo.
ClearDataCategories		Borra los textos establecidos por **SetDataCategories**.
ZoomRange(aMin, aMax: Double)		Restringe el rango visible a **[aMin, aMax]**.
ResetZoom		Borra un rango establecido por **ZoomRange**.
Zoomed	Boolean (solo lectura)	True mientras hay un rango de zoom activo.
VisibleMin, VisibleMax	Double (solo lectura)	El rango actualmente en vigor: el rango de zoom, o si no, el rango manual o de datos.
Calculate(aCanvas: TsgcUICanvas; aPixelLength: Double; aScale: Double = 1)		Calcula las marcas para un trazado de **aPixelLength** píxeles a lo largo de este eje. **aCanvas** mide las etiquetas para el adelgazamiento y puede ser nil.
TickCount, TickValue[Index], TickLabel[Index]	Integer, Double, string	Las marcas calculadas por el último **Calculate**. **TickLabel** está vacío para una etiqueta eliminada por el adelgazamiento.
Step, DateTimeUnit	Double, TsgcChartDateTimeUnit	El espaciado entre marcas y, para un eje `catDateTime`, la unidad a la que se redondeó ese espaciado.
ValueToPixel(aValue, aStart, aLength: Double)	Double	Asigna un valor de datos a una posición de píxel de dispositivo.
PixelToValue(aPixel, aStart, aLength: Double)	Double	La inversa de **ValueToPixel**.
MeasureSize(aCanvas: TsgcUICanvas; aScale: Double = 1)	Double	Grosor que este eje necesita fuera del trazado (marcas, etiquetas, título); 0 cuando no es **Visible**.
Draw, DrawGrid		Dibuja el eje, o sus líneas de cuadrícula, para el último **Calculate**.

Miembros de TsgcChartAxes

Miembro	Descripción
Add: TsgcChartAxis	Añade un nuevo eje y lo devuelve.
Find(const aName: string): TsgcChartAxis	Primer eje con este Name (sin distinguir mayúsculas y minúsculas), nil cuando no hay ninguno.
Items[Index: Integer]: TsgcChartAxis (default)	Eje en Index.

Rango automático y zoom

Mientras **Automatic** es True, el gráfico proporciona al eje los extremos de datos combinados de cada serie enlazada a él en cada renderizado, mediante **SetDataRange**; un eje automático amplía ese rango hasta valores redondos de marca, salvo que el gráfico llame a **SetDataRange** con **aExact** en True, lo que hace para **TsgcChart.TimeWindow**. Establecer **Automatic** en False e indicar **Minimum** / **Maximum** fija el rango en su lugar. **ZoomRange** y **ResetZoom** se aplican por encima de cualquiera de los dos, para el zoom interactivo; **VisibleMin** / **VisibleMax** informan del rango que está realmente en vigor.

Formato de etiqueta

Para un eje `catDateTime`, **LabelFormat** utiliza letras de fecha y hora independientes de la configuración regional, sin depender de la configuración regional en tiempo de ejecución: `yyyy`, `yy`, `mmmm`, `mmm`, `mm`, `m`, `dd`, `d`, `hh`, `h`, `nn`, `n`, `ss`, `s`, `zzz`. El texto entre comillas se copia tal cual. `mm` es siempre el mes; utilice `nn` para los minutos. Si se deja vacío, el eje elige un formato a partir de **DateTimeUnit** del espaciado de marcas automáticamente.

Categories

Un eje `catCategory` coloca una marca por cada entrada de **Categories**, en el orden de la lista, en lugar de calcular marcas numéricas; las series enlazadas a él se dibujan en posiciones enteras que coinciden con esas entradas. Cuando **Categories** está vacío, una serie puede proporcionar los textos en tiempo de ejecución: un mapa de calor etiqueta así sus líneas y sus celdas.

Varios ejes de valores

Un gráfico no está limitado a un eje X y un eje Y: añadir un segundo eje con la misma **Position** que uno existente, y apuntar algunas series hacia él mediante **AxisX** / **AxisY**, da a esas series su propia escala mientras el resto sigue utilizando el primer eje. Una serie que deja **AxisX** o **AxisY** vacío se enlaza al primer eje inferior y al primer eje izquierdo, respectivamente.

Ejemplo: Un eje inferior de fecha y hora y un eje izquierdo numérico

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vAxisX, vAxisY: TsgcChartAxis;
  vSeries: TsgcChartSeries_Line;
  vNow: TDateTime;
```

```
  i: Integer;
  vRaster: TsgcUICanvas_Raster;
begin
  vChart := TsgcChart.Create(nil);
  try
    vAxisX := vChart.Axes.Add;
    vAxisX.Name := 'time';
    vAxisX.Position := capBottom;
    vAxisX.AxisType := catDateTime;
    vAxisX.LabelFormat := 'dd/mm hh:nn';

    vAxisY := vChart.Axes.Add;
    vAxisY.Name := 'value';
    vAxisY.Position := capLeft;
    vAxisY.LabelFormat := '0.00';

    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Chart := vChart;
      vSeries.AxisX := 'time';
      vSeries.AxisY := 'value';
      vNow := Now;
      for i := 0 to 59 do
        vSeries.AddXY(vNow + (i / 1440), 50 + Sin(i / 5) * 10);

      { Step and VisibleMin / VisibleMax are only computed by Render (it
        calls Calculate on every axis), so read them after rendering }
      vRaster := TsgcUICanvas_Raster.Create(700, 350);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 700, 350));
        vRaster.SaveToPNGFile('axes.png');

        Writeln('X ticks: ', vAxisX.TickCount, ', unit: ', Ord(vAxisX.DateTimeUnit));
        Writeln('Y visible range: ', vAxisY.VisibleMin, ' .. ', vAxisY.VisibleMax);
      finally
        vRaster.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChart
- TsgcChartSeries

Título, Leyenda y Márgenes

Unidad: sgcCharts_Chart
Tres clases `TPersistent` pequeñas contienen la decoración que rodea al gráfico: **TsgcChartTitle** (utilizada tanto para TsgcChart.**Title** como para **Subtitle**), **TsgcChartLegend** (**TsgcChart.Legend**) y **TsgcChartMargins** (**TsgcChart.Margins**). Las tres generan una notificación interna de cambio que invalida el gráfico, de modo que editarlas en tiempo de ejecución repinta automáticamente.

Propiedades de TsgcChartTitle

Propiedad	Tipo	Descripción
Text	string	El texto del título. Un **Text** vacío no reserva espacio; el título se omite cuando está vacío o no es **Visible**.
TextStyle	TsgcUITextStyle	Tamaño de fuente, negrita y color. Véase más abajo.
Visible	Boolean	Predeterminado True.
Alignment	TAlignment	Predeterminado `taCenter`.

Propiedades de TsgcChartLegend

Propiedad	Tipo	Descripción
Visible	Boolean	Predeterminado True.
Position	TsgcChartLegendPosition	`clpTop` (predeterminado), `clpBottom`, `clpLeft` o `clpRight`.
TextStyle	TsgcUITextStyle	Fuente de las entradas de la leyenda.
BackgroundColor	TsgcUIColor	Predeterminado 0 (alfa 0, transparente).
Padding	Integer	Predeterminado 4, píxeles de dispositivo a escala 1.

Una serie aparece en la leyenda cuando su propio **ShowInLegend** es True (véase TsgcChartSeries); el texto de la leyenda es el **LegendText** de la serie.

Propiedades de TsgcChartMargins

Propiedad	Tipo	Descripción
Left, Top, Right, Bottom	Integer	Margen exterior en píxeles de dispositivo a escala 1, alrededor de todo el gráfico, fuera del título, la leyenda y los ejes. Predeterminado 10 en cada lado.

Propiedades de TsgcUITextStyle

Unidad: sgcCharts_Types. Utilizada por **TsgcChartTitle.TextStyle**, **TsgcChartLegend.TextStyle** y **TsgcChartAxis.TitleStyle** / **LabelStyle**.

Propiedad	Tipo	Descripción
Size	Double	Tamaño de fuente en puntos. Predeterminado 11.
Bold	Boolean	Predeterminado False.
Color	TsgcUIColor	

ToFontStyle(aScale: Double = 1) devuelve el **TsgcUIFontStyle** correspondiente para una llamada de lienzo, con **Size** multiplicado por **aScale**.

Elementos de leyenda y barras de color

Una serie puede añadir varias entradas a la leyenda, y las filas de la leyenda toman la altura de su entrada más alta. Un gráfico circular añade una entrada por cada sector dibujado, con Otros al final; un embudo una por cada etapa dibujada; un treemap en modo `ctcGroup` una por cada grupo de nivel superior. Un mapa de calor, y un treemap en modo `ctcValue`, añaden una barra de color con el mínimo y el máximo de su escala de color, horizontal o vertical según la posición de la leyenda. **ShowInLegend** en False oculta todas las entradas de una serie, y una serie omitida por su área de trazado no muestra ninguna.

Ejemplo: Título, subtítulo, posición de la leyenda y márgenes

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  i: Integer;
  vRaster: TsgcUICanvas_Raster;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Title.Text := 'Server Load';
    vChart.Subtitle.Text := 'last 24 hours';
    vChart.Subtitle.TextStyle.Size := 9;

    vChart.Legend.Position := clpBottom;
    vChart.Legend.Padding := 6;

    vChart.Margins.Left := 16;
    vChart.Margins.Top := 16;
    vChart.Margins.Right := 16;
    vChart.Margins.Bottom := 16;

    vChart.Axes.Add.Position := capBottom;
    vChart.Axes.Add.Position := capLeft;

    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Chart := vChart;
      vSeries.Title := 'CPU %';
```

```
      for i := 0 to 49 do
        vSeries.AddXY(i, 40 + Sin(i / 6) * 20);

      vRaster := TsgcUICanvas_Raster.Create(640, 360);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 640, 360));
        vRaster.SaveToPNGFile('title_legend.png');
      finally
        vRaster.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChart
- TsgcChartSeries

Temas y Paleta

Unidad: sgcCharts_Chart (TsgcChartPalette), sgcCharts_Types (TsgcUITheme)
TsgcChart.**Theme** elige en un solo paso el fondo, el texto, la cuadrícula y los colores de línea de eje integrados, así como la paleta de series predeterminada; **Palette** es la lista de 8 colores que las series utilizan automáticamente por índice.

TsgcUITheme

Valor	Descripción
utLight	Predeterminado. Fondo claro, texto oscuro, cuadrícula clara.
utDark	Fondo oscuro, texto claro, cuadrícula oscura.
utCustom	Deja **BackgroundColor**, **PlotBackgroundColor** y **Palette** exactamente como están; el gráfico no aplica ningún color integrado.

`utLight` y `utDark` llevan cada uno su propia **Palette** predeterminada (8 colores, utilizados en el orden de índice de las series). Establecer **BackgroundColor**, **PlotBackgroundColor** o cualquier color de **Palette** directamente cambia **Theme** a `utCustom` automáticamente, salvo que el gráfico se esté cargando desde un flujo; a partir de ese momento, volver a establecer **Theme** en `utLight` o `utDark` vuelve a aplicar los colores de ese tema y sobrescribe los personalizados.

Propiedades de TsgcChartPalette

Propiedad	Tipo	Descripción
Colors[Index: Integer] (default)	TsgcUIColor	Index se toma módulo 8, por lo que nunca genera un error para un índice de serie fuera de rango.
Color1 .. Color8	TsgcUIColor	Los 8 colores como propiedades publicadas, índice 0 a 7.

Métodos de TsgcChartPalette

Método	Descripción
LoadColors(const aColors: array of TsgcUIColor)	Copia hasta 8 colores de **aColors** en **Color1**.. **Color8** en orden.

Una serie con **Color** = 0 (el valor predeterminado) toma **Palette.Colors[Series.Index]** a través de **EffectiveColor**; véase TsgcChartSeries.

Ejemplo: Tema oscuro y, a continuación, una paleta personalizada

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
```

```
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vSeries1, vSeries2: TsgcChartSeries_Line;
  i: Integer;
  vRaster: TsgcUICanvas_Raster;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Theme := utDark;
    vChart.Axes.Add.Position := capBottom;
    vChart.Axes.Add.Position := capLeft;

    vSeries1 := TsgcChartSeries_Line.Create(nil);
    vSeries2 := TsgcChartSeries_Line.Create(nil);
    try
      vSeries1.Chart := vChart;
      vSeries1.Title := 'A';
      vSeries2.Chart := vChart;
      vSeries2.Title := 'B';
      for i := 0 to 39 do
      begin
        vSeries1.AddXY(i, Sin(i / 5) * 20 + 50);
        vSeries2.AddXY(i, Cos(i / 5) * 20 + 50);
      end;

      { override the first automatic colour; Theme switches to utCustom }
      vChart.Palette.Color1 := $FFFFA500;
      Writeln('theme after palette edit: ', Ord(vChart.Theme));

      vRaster := TsgcUICanvas_Raster.Create(640, 320);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 640, 320));
        vRaster.SaveToPNGFile('theme_palette.png');
      finally
        vRaster.Free;
      end;
    finally
      vSeries1.Free;
      vSeries2.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChart
- TsgcChartSeries

Áreas de trazado

Unidad: sgcCharts_Chart
Un área de trazado es el trazado de un gráfico, o un panel visible cuando el gráfico tiene paneles. En cada renderizado el gráfico asigna a cada área un modo, calculado a partir de las series visibles enlazadas a ella:

Modo	Series	Qué dibuja el área
`cpmXY`	Línea, área, barras, dispersión, velas, mapa de calor y las demás series XY	Ejes, cuadrícula y cada serie XY mediante **Draw**.
`cpmPolar`	TsgcChartSeries_Radar	Una cuadrícula polar; el primer radar dibuja sobre ella todos los radares del área.
`cpmCell`	TsgcChartSeries_Pie, TsgcChartSeries_Funnel, TsgcChartSeries_Treemap	Una cuadrícula de celdas, una por serie, cada una con una banda de título; sin ejes.

La regla de mezcla

XY prevalece siempre que hay una serie XY visible en el área. En caso contrario, la primera serie visible decide entre polar y celda. Las series que no coinciden con el modo de su área se omiten: no se dibujan, no reciben impactos y no aparecen en la leyenda. Para mostrar un gráfico circular junto a un gráfico de líneas, dé al gráfico dos paneles y enlace cada serie al eje de su panel, o use dos gráficos.

Miembros de área de TsgcChart

Método	Devuelve	Descripción
AreaCount	Integer (solo lectura)	Áreas de trazado del último **Render**: 1 para el trazado, o una por panel visible en el orden de la colección.
AreaRect(aIndex: Integer)	TsgcUIRect	Rectángulo de un área, vacío cuando **aIndex** está fuera de rango.
AreaMode(aIndex: Integer)	TsgcChartPlotMode	Modo de un área, `cpmXY` cuando **aIndex** está fuera de rango.
AreaAt(X, Y: Double)	Integer	Índice del área bajo un píxel del gráfico, -1 si no hay ninguna.
SeriesAreaRect(aSeries: TsgcChartSeries)	TsgcUIRect	La celda de una serie de celda dibujada, el área de una serie polar dibujada, el rectángulo de trazado del panel de una serie XY, vacío para cualquier otra serie.
IsSeriesDrawn(aSeries: TsgcChartSeries)	Boolean	True cuando el último **Render** dibujó la serie: visible, del modo de su área, y para una serie XY con sus dos ejes.

Las celdas se disponen como una cuadrícula que llena el área. Con varias celdas, cada una empieza bajo una banda de título que contiene el **Title** de la serie. La interacción dirige el desplazamiento del ratón, el clic y las informaciones sobre herramientas a las series de todos los modos, pero el zoom, el desplazamiento y la mira solo funcionan en áreas XY. El autor de una serie implementa los ganchos enumerados en TsgcChartSeries.

Ejemplo: dos series de celda y después una serie XY

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Category,
  sgcCharts_Series_Pie, sgcCharts_Series_Funnel;

const
  CS_MODES: array [TsgcChartPlotMode] of string = ('XY', 'polar', 'cell');

var
  vChart: TsgcChart;
  vPie: TsgcChartSeries_Pie;
  vFunnel: TsgcChartSeries_Funnel;
  vLine: TsgcChartSeries_Line;
  vRaster: TsgcUICanvas_Raster;
  vRect: TsgcUIRect;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  vRaster := TsgcUICanvas_Raster.Create(800, 400);
  vPie := TsgcChartSeries_Pie.Create(nil);
  vFunnel := TsgcChartSeries_Funnel.Create(nil);
  vLine := TsgcChartSeries_Line.Create(nil);
  try
    vPie.Chart := vChart;
    vPie.Title := 'Share';
    vPie.SetValue('Web', 55);
    vPie.SetValue('Store', 30);
    vPie.SetValue('Phone', 15);

    vFunnel.Chart := vChart;
    vFunnel.Title := 'Pipeline';
    vFunnel.SetValue('Leads', 900);
    vFunnel.SetValue('Offers', 420);
    vFunnel.SetValue('Orders', 160);

    // two cell series: the plot area is split into two cells
    vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
    vRaster.SaveToPNGFile('cells.png');
    Writeln('areas: ', vChart.AreaCount, ', mode: ',
      CS_MODES[vChart.AreaMode(0)]);
    vRect := vChart.SeriesAreaRect(vFunnel);
    Writeln('funnel cell: ', FormatFloat('0', vRect.Left), ', ',
      FormatFloat('0', vRect.Top), ' to ', FormatFloat('0', vRect.Right),
      ', ', FormatFloat('0', vRect.Bottom));

    // one visible XY series turns the area into an XY area:
    // the pie and the funnel are skipped
    vLine.Chart := vChart;
    vLine.Title := 'Orders per day';
    for i := 0 to 9 do
      vLine.AddXY(i, 10 + i * i);
    vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
    vRaster.SaveToPNGFile('xy.png');
    Writeln('mode: ', CS_MODES[vChart.AreaMode(0)], ', pie drawn: ',
      vChart.IsSeriesDrawn(vPie), ', line drawn: ',
      vChart.IsSeriesDrawn(vLine));
  finally
    vLine.Free;
    vFunnel.Free;
    vPie.Free;
    vRaster.Free;
    vChart.Free;
```

```
  end;
end.
```

Véase también

- TsgcChart
- TsgcChartSeries
- Panes
- TsgcChartInteraction

Escala de color

Unidad: sgcCharts_ColorScale
TsgcChartColorScale asigna un color a un valor mediante una tabla de búsqueda (LUT) de 256 entradas construida a partir de dos o tres paradas de color. Es la propiedad **ColorScale** de TsgcChartSeries_Heatmap y de TsgcChartSeries_Treemap en modo `ctcValue`, y la leyenda la muestra como una barra de color con el mínimo y el máximo.

Propiedades

Propiedad	Tipo	Descripción
MinColor	TsgcUIColor	Color del mínimo; 0 (predeterminado) usa el fondo del trazado del tema.
MidColor	TsgcUIColor	Color del centro del rango; 0 (predeterminado) significa dos paradas.
MaxColor	TsgcUIColor	Color del máximo; 0 (predeterminado) usa el color de la serie, o si no el primer color de la paleta.
Steps	Integer	0 (predeterminado) es una escala continua; de 2 a 32 es una paleta escalonada con ese número de bandas.
Automatic	Boolean	Valor predeterminado True: el rango es el mínimo y el máximo de las celdas o mosaicos visibles. False usa **Minimum** y **Maximum**.
Minimum, Maximum	Double	El rango fijo. Valores predeterminados 0 y 100.
NullColor	TsgcUIColor	Color de las celdas NaN; 0 (predeterminado) las deja sin pintar.
RangeMin, RangeMax	Double (solo lectura)	El rango establecido por el último renderizado.
LUT[aIndex]	TsgcUIColor (solo lectura)	Una entrada de la tabla de búsqueda, de 0 a 255.
OnChange	TNotifyEvent	Se dispara cuando cambia una propiedad; la serie propietaria se repinta.

Métodos

Método	Devuelve	Descripción
Prepare(aDefaultMinColor, aDefaultMaxColor: TsgcUIColor; const aDataMin, aDataMax: Double)		La llama la serie antes de dibujar: resuelve los colores cero, reconstruye la LUT cuando ha cambiado una parada o **Steps** y establece el rango. Un rango de ancho cero asigna todos los valores a la entrada 128.

IndexOf(const aValue: Double)	Integer	Entrada de la LUT de un valor, de 0 a 255; -1 para NaN.
ColorOf(const aValue: Double)	TsgcUIColor	Color de la LUT de un valor; **NullColor** para NaN.
DrawBar, MeasureColorBar, DrawColorBar		Dibujan la barra y la barra de color de la leyenda con sus etiquetas; los usan los elementos de leyenda de las series.

Ejemplo: una escala escalonada de semáforo sobre un rango fijo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_ColorScale, sgcCharts_Series_Heatmap;

var
  vChart: TsgcChart;
  vHeat: TsgcChartSeries_Heatmap;
  vScale: TsgcChartColorScale;
  vRaster: TsgcUICanvas_Raster;
  vValues: array [0 .. 9] of Double;
  vHour, j: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vHeat := TsgcChartSeries_Heatmap.Create(nil);
    try
      vHeat.Chart := vChart;
      vHeat.Title := 'Load %';
      // three colour stops and five steps on a fixed 0..100 range
      vScale := vHeat.ColorScale;
      vScale.MinColor := $FF2E7D32;
      vScale.MidColor := $FFFDD835;
      vScale.MaxColor := $FFC62828;
      vScale.Steps := 5;
      vScale.Automatic := False;
      vScale.Minimum := 0;
      vScale.Maximum := 100;
      vScale.NullColor := $FFBDBDBD;

      // one line per hour (X), ten servers per line (Y)
      vHeat.BeginUpdate;
      try
        for vHour := 0 to 23 do
        begin
          for j := 0 to High(vValues) do
            vValues[j] := (vHour * 7 + j * 13) mod 100;
          vHeat.AppendLine(vHour, vValues);
        end;
      finally
        vHeat.EndUpdate;
      end;

      vRaster := TsgcUICanvas_Raster.Create(800, 360);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 360));
        vRaster.SaveToPNGFile('scale.png');
      finally
        vRaster.Free;
      end;
```

```
      // the range and the colours of the last render
      Writeln('range: ', FormatFloat('0', vScale.RangeMin), ' to ',
        FormatFloat('0', vScale.RangeMax));
      Writeln('42 is LUT entry ', vScale.IndexOf(42), ', colour $',
        IntToHex(vScale.ColorOf(42), 8));
    finally
      vHeat.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChartSeries_Heatmap
- TsgcChartSeries_Treemap
- Título, Leyenda y Márgenes
- Temas y Paleta

TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine

Unidad: sgcCharts_Series
Estas tres series descienden de **TsgcChartSeries_XY**, la base común de toda serie X/Y (línea, línea rápida, línea escalonada, área, barra y dispersión, véase TsgcChartSeries para las propiedades que comparten todas las series). TsgcChartSeries_XY posee el búfer de puntos y los métodos que lo alimentan; **TsgcChartSeries_Line** dibuja una polilínea con un patrón de trazos opcional y marcadores de punto; **TsgcChartSeries_FastLine** es una variante más fina pensada para grandes volúmenes de datos en vivo; **TsgcChartSeries_StepLine** desciende de TsgcChartSeries_Line y dibuja una línea escalonada en lugar de segmentos rectos.

TsgcChartSeries_XY

TsgcChartSeries_XY posee un **Buffer** de puntos X/Y e implementa los métodos de datos que comparten todas las series de línea, área, línea escalonada, barra y dispersión. No es abstracta, pero normalmente se usa a través de una de sus descendientes en lugar de por sí sola.

Propiedad	Tipo	Descripción
Buffer	TsgcChartBuffer (solo lectura)	El búfer circular subyacente. Véase TsgcChartBuffer y TsgcChartOHLCBuffer.
Capacity	Integer	Número máximo de puntos que contiene el búfer. Valor predeterminado 100000.
BufferMode	TsgcChartBufferMode	`cbmOverwrite` (valor predeterminado) descarta el punto más antiguo cuando el búfer está lleno; `cbmGrow` duplica la capacidad en su lugar.

Métodos

Método	Devuelve	Descripción
AddXY(const aX, aY: Double)		Añade un punto y llama a **Changed**.
AddY(const aY: Double)		Añade un punto en X igual a la X del punto más reciente más 1 (0 para un búfer vacío) y llama a **Changed**. Útil cuando X es un índice consecutivo en lugar de una coordenada real.
Clear		Vacía el búfer y llama a **Changed**.

TsgcChartSeries_Line

TsgcChartSeries_Line dibuja los puntos del búfer como una polilínea, con un patrón de trazos opcional y marcadores de punto opcionales.

Propiedad	Tipo	Descripción
LineWidth	Double	Píxeles de dispositivo a escala 1. Valor predeterminado 2.
Dash	TsgcChart-DashStyle	`cdsSolid`, `cdsDash`, `cdsDot` o `cdsDashDot`. Valor predeterminado `cdsSolid`.
Markers	TsgcChartSe-riesMarkers	Oculto de forma predeterminada (**Visible** es False). Véase TsgcChartSe-riesMarkers a continuación.

TsgcChartSeries_FastLine

TsgcChartSeries_FastLine es una línea fina pensada para grandes volúmenes de datos en vivo: no tiene marcadores ni patrón de trazos, y sus segmentos se unen con una unión biselada en lugar de la unión redondeada que usa TsgcChartSeries_Line, lo que la hace económica de redibujar a una frecuencia de actualización alta.

Propiedad	Tipo	Descripción
LineWidth	Double	Píxeles de dispositivo a escala 1. Valor predeterminado 1.

TsgcChartSeries_StepLine

TsgcChartSeries_StepLine desciende de TsgcChartSeries_Line, por lo que también tiene **LineWidth**, **Dash** y **Markers**, pero conecta los puntos consecutivos con un segmento horizontal y uno vertical en lugar de una diagonal recta.

Propiedad	Tipo	Descripción
StepMode	TsgcChartStep-Mode	`csmBefore`, `csmAfter` o `csmCenter`. Valor predeterminado `csmAf-ter`.

Con **csmAfter**, un valor se mantiene hasta que se alcanza el punto siguiente, de modo que el escalón ocurre justo en cada punto. Con **csmBefore**, el valor se alcanza en el punto anterior, de modo que el escalón ocurre justo antes de él. Con **csmCenter**, el escalón ocurre a mitad de camino entre los dos puntos.

TsgcChartSeriesMarkers

TsgcChartSeriesMarkers es el pequeño objeto persistente expuesto por la propiedad **Markers** de TsgcChartSeries_Line y de TsgcChartSeries_Scatter. Dibuja una forma centrada en cada punto visible.

Propie-dad	Tipo	Descripción
Visible	Boolean	Indica si se dibujan los marcadores. El valor predeterminado depende de la serie propietaria: False para TsgcChartSeries_Line y sus descendientes, True para TsgcChartSeries_Scatter.
Shape	Tsgc-ChartMar-kerShape	`cmsSquare`, `cmsCircle`, `cmsTriangle` o `cmsDiamond`. Valor predeterminado `cmsCircle`.
Size	Integer	Diámetro en píxeles de dispositivo a escala 1. Valor predeterminado 6.

Ejemplo: series de línea, línea rápida y línea escalonada en un mismo gráfico

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vLine: TsgcChartSeries_Line;
  vFast: TsgcChartSeries_FastLine;
  vStep: TsgcChartSeries_StepLine;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Axes.Add.Name := 'x';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'y';
    vChart.Axes[1].Position := capLeft;

    vLine := TsgcChartSeries_Line.Create(nil);
    vFast := TsgcChartSeries_FastLine.Create(nil);
    vStep := TsgcChartSeries_StepLine.Create(nil);
    try
      vLine.Chart := vChart;
      vLine.AxisX := 'x';
      vLine.AxisY := 'y';
      vLine.Dash := cdsDash;
      vLine.Markers.Visible := True;
      vLine.Markers.Shape := cmsDiamond;
      for i := 0 to 29 do
        vLine.AddXY(i, Sin(i / 5) * 40 + 60);

      vFast.Chart := vChart;
      vFast.AxisX := 'x';
      vFast.AxisY := 'y';
      vFast.LineWidth := 3;
      for i := 0 to 29 do
        vFast.AddXY(i, Cos(i / 5) * 20 + 20);

      vStep.Chart := vChart;
      vStep.AxisX := 'x';
      vStep.AxisY := 'y';
      vStep.StepMode := csmCenter;
      for i := 0 to 9 do
        vStep.AddXY(i * 3, (i mod 2) * 30);

      Writeln('line dash: ', Ord(vLine.Dash));
      Writeln('step mode: ', Ord(vStep.StepMode));

      vRaster := TsgcUICanvas_Raster.Create(600, 300);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 600, 300));
        vRaster.SaveToPNGFile('line.png');
      finally
        vRaster.Free;
      end;
    finally
      vLine.Free;
      vFast.Free;
      vStep.Free;
    end;
  finally
```

```
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChart
- TsgcChartSeries
- TsgcChartSeries_Area
- TsgcChartSeries_Scatter

TsgcChartSeries_Area

Unidad: sgcCharts_Series
TsgcChartSeries_Area desciende de TsgcChartSeries_Line, por lo que dibuja la misma polilínea (con **LineWidth**, **Dash** y **Markers**) y además rellena el área entre esa línea y un valor de línea base.

Propiedades

Propiedad	Tipo	Descripción
Baseline	Double	El relleno va desde la línea hacia abajo (o hacia arriba, cuando la línea está por debajo) hasta este valor. Valor predeterminado 0.
FillOpacity	Double	0..1. Valor predeterminado 0.35.
Gradient	Boolean	False (valor predeterminado) rellena con un color plano en **FillOpacity**. True rellena con un degradado vertical que va desde el color de relleno en **FillOpacity** hasta ser totalmente transparente en **Baseline**.

Ejemplo: una serie de área con una línea base elevada y un relleno degradado

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vArea: TsgcChartSeries_Area;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Axes.Add.Name := 'x';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'y';
    vChart.Axes[1].Position := capLeft;

    vArea := TsgcChartSeries_Area.Create(nil);
    try
      vArea.Chart := vChart;
      vArea.AxisX := 'x';
      vArea.AxisY := 'y';
      vArea.Baseline := 10;
      vArea.FillOpacity := 0.5;
      vArea.Gradient := True;

      for i := 0 to 59 do
        vArea.AddXY(i, Sin(i / 8) * 25 + 40);

      Writeln('baseline: ', vArea.Baseline: 0: 1);
      Writeln('fill opacity: ', vArea.FillOpacity: 0: 2);

      vRaster := TsgcUICanvas_Raster.Create(600, 300);
```

```
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 600, 300));
        vRaster.SaveToPNGFile('area.png');
      finally
        vRaster.Free;
      end;
    finally
      vArea.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChart
- TsgcChartSeries
- TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine

TsgcChartSeries_Bar

Unidad: sgcCharts_Series
TsgcChartSeries_Bar desciende de **TsgcChartSeries_XY** (véase TsgcChartSeries_XY para **Buffer**, **AddXY**, **AddY**, **Clear**, **Capacity** y **BufferMode**) y dibuja una barra por punto.

Propiedades

Propiedad	Tipo	Descripción
Decimation	TsgcChartDecimation	Valor predeterminado `dmNone` para una serie de barras, a diferencia del valor predeterminado `dmMinMax` de la clase base TsgcChartSeries: las barras están pensadas para mostrarse una por columna de píxeles o más ancha, no para decimarse. Véase Decimación.
Orientation	TsgcChartBarOrientation	`cboVertical` (valor predeterminado) o `cboHorizontal`. Una serie de barras horizontal asigna su X (posición) al eje vertical y su Y (valor) al eje horizontal, de modo que las barras crecen hacia los lados en lugar de hacia arriba.
BarWidthPercent	Integer	Proporción del hueco X ocupada por la barra, o por el grupo de barras, 1..100. Valor predeterminado 70.
Grouping	TsgcChartBarGrouping	`cbgNone` (valor predeterminado), `cbgGrouped` o `cbgStacked`. Véase a continuación.

Agrupación y apilamiento

Con **cbgGrouped**, las series de barras visibles que comparten los mismos ejes, orientación y agrupación se colocan una junto a otra dentro de cada hueco X. Con **cbgStacked**, esas series se apilan en su lugar, en el orden de las series: los valores positivos se apilan en el lado positivo y los negativos en el lado negativo, de modo que una serie con un valor negativo en un X determinado no empuja hacia arriba la pila positiva. **cbgNone** dibuja cada serie de barras por separado, superponiéndose a las demás en el mismo X.

Ejemplo: dos series de barras agrupadas

```
uses
  SysUtils,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vBar1, vBar2: TsgcChartSeries_Bar;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Axes.Add.Name := 'x';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'y';
    vChart.Axes[1].Position := capLeft;
```

```
    vBar1 := TsgcChartSeries_Bar.Create(nil);
    vBar2 := TsgcChartSeries_Bar.Create(nil);
    try
      vBar1.Chart := vChart;
      vBar1.AxisX := 'x';
      vBar1.AxisY := 'y';
      vBar1.Title := 'Budget';
      vBar1.Orientation := cboVertical;
      vBar1.Grouping := cbgGrouped;
      vBar1.BarWidthPercent := 80;

      vBar2.Chart := vChart;
      vBar2.AxisX := 'x';
      vBar2.AxisY := 'y';
      vBar2.Title := 'Actual';
      vBar2.Orientation := cboVertical;
      vBar2.Grouping := cbgGrouped;
      vBar2.BarWidthPercent := 80;

      for i := 0 to 5 do
      begin
        vBar1.AddXY(i, 40 + i * 5);
        vBar2.AddXY(i, 35 + i * 6);
      end;

      Writeln('bar1 orientation: ', Ord(vBar1.Orientation));
      Writeln('bar grouping: ', Ord(vBar1.Grouping));

      vRaster := TsgcUICanvas_Raster.Create(600, 300);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 600, 300));
        vRaster.SaveToPNGFile('bars.png');
      finally
        vRaster.Free;
      end;
    finally
      vBar1.Free;
      vBar2.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChart
- TsgcChartSeries
- Decimación

TsgcChartSeries_Scatter

Unidad: sgcCharts_Series
TsgcChartSeries_Scatter desciende de **TsgcChartSeries_XY** (véase TsgcChartSeries_XY para **Buffer**, **AddXY**, **AddY**, **Clear**, **Capacity** y **BufferMode**) y dibuja sus puntos solo como marcadores, sin línea de conexión.

Propiedades

Propiedad	Tipo	Descripción
Decimation	TsgcChartDecimation	Valor predeterminado `dmNone` para una serie de dispersión, a diferencia del valor predeterminado `dmMinMax` de la clase base TsgcChartSeries: un diagrama de dispersión no tiene línea que mantener visualmente exacta, así que se conserva cada punto. Véase Decimación.
Markers	TsgcChartSeriesMarkers	Visible de forma predeterminada (a diferencia del Markers de la serie de línea). Véase TsgcChartSeriesMarkers para **Shape** y **Size**; establecer **Visible** en False oculta los puntos.

Ejemplo: una serie de dispersión con marcadores triangulares

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vScatter: TsgcChartSeries_Scatter;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Axes.Add.Name := 'x';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'y';
    vChart.Axes[1].Position := capLeft;

    vScatter := TsgcChartSeries_Scatter.Create(nil);
    try
      vScatter.Chart := vChart;
      vScatter.AxisX := 'x';
      vScatter.AxisY := 'y';
      vScatter.Markers.Shape := cmsTriangle;
      vScatter.Markers.Size := 8;

      for i := 0 to 49 do
        vScatter.AddXY(i, Sin(i / 3) * 30 + (i mod 7));

      Writeln('decimation: ', Ord(vScatter.Decimation));
      Writeln('markers visible: ', vScatter.Markers.Visible);

      vRaster := TsgcUICanvas_Raster.Create(600, 300);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 600, 300));
```

```
        vRaster.SaveToPNGFile('scatter.png');
      finally
        vRaster.Free;
      end;
    finally
      vScatter.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChart
- TsgcChartSeries
- TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine
- Decimación

TsgcChartSeries_Candlestick, TsgcChartSeries_OHLC

Unidad: sgcCharts_Series
Estas dos series descienden de **TsgcChartSeries_OHLCBase**, la base común de toda serie de apertura/máximo/mínimo/cierre. TsgcChartSeries_OHLCBase posee un búfer OHLC en lugar del búfer X/Y que usan las series de la familia de líneas, e implementa los métodos que lo alimentan; **TsgcChartSeries_Candlestick** dibuja cada barra como un cuerpo de vela relleno u hueco con mechas, y **TsgcChartSeries_OHLC** dibuja cada barra como una línea vertical con marcas de apertura y cierre.

TsgcChartSeries_OHLCBase

Propiedad	Tipo	Descripción
OHLCBuffer	TsgcChartOHLCBuffer (solo lectura)	El búfer circular subyacente de velas. Véase TsgcChartBuffer y TsgcChartOHLCBuffer.
Capacity	Integer	Número máximo de velas que contiene el búfer. Valor predeterminado 100000.
BufferMode	TsgcChartBufferMode	`cbmOverwrite` (valor predeterminado) descarta la vela más antigua cuando el búfer está lleno; `cbmGrow` duplica la capacidad en su lugar.
UpColor	TsgcUIColor	Color usado cuando el cierre de una vela es mayor o igual que su apertura. Valor predeterminado un verde, $FF1BAF7A.
DownColor	TsgcUIColor	Color usado cuando el cierre de una vela es menor que su apertura. Valor predeterminado un rojo, $FFE34948.

Métodos

Método	Devuelve	Descripción
AddOHLC(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)		Añade una vela y llama a **Changed**.
UpdateLastOHLC(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)		Sustituye la vela más reciente cuando su X es igual a **aX**, o la añade en caso contrario, y llama a **Changed**. Pensado para una fuente en vivo que sigue actualizando la vela en curso hasta que su periodo se cierra.
Clear		Vacía el búfer y llama a **Changed**.

TsgcChartSeries_Candlestick

TsgcChartSeries_Candlestick dibuja cada vela como un cuerpo entre la apertura y el cierre, coloreado con **UpColor** o **DownColor**, más una mecha desde el máximo hasta el mínimo.

Propiedad	Tipo	Descripción
WickColor	TsgcUI-Color	Valor predeterminado 0, lo que significa que la mecha toma el color del cuerpo de la vela. Un valor distinto de cero lo sobrescribe para todas las velas.
BodyWidth-Percent	Integer	Proporción del hueco de la vela ocupada por el cuerpo, 1..100. Valor predeterminado 70.
HollowUp	Boolean	Valor predeterminado False. True dibuja las velas alcistas (cierre >= apertura) como un contorno en **UpColor** en lugar de un cuerpo relleno.

TsgcChartSeries_OHLC

TsgcChartSeries_OHLC dibuja cada barra como una línea vertical desde el máximo hasta el mínimo, con una marca corta a la izquierda en la apertura y una marca corta a la derecha en el cierre, coloreada con **UpColor** o **DownColor**.

Propiedad	Tipo	Descripción
TickWidthPer-cent	Inte-ger	Proporción del hueco de la vela ocupada por las marcas de apertura y cierre, 1..100. Valor predeterminado 70.
LineWidth	Dou-ble	Píxeles de dispositivo a escala 1. Valor predeterminado 1.

Ejemplo: una serie de velas y una serie OHLC

```
uses
  SysUtils,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vCandle: TsgcChartSeries_Candlestick;
  vOHLC: TsgcChartSeries_OHLC;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
  vOpen, vClose: Double;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Axes.Add.Name := 'x';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'y';
    vChart.Axes[1].Position := capLeft;

    vCandle := TsgcChartSeries_Candlestick.Create(nil);
    vOHLC := TsgcChartSeries_OHLC.Create(nil);
    try
      vCandle.Chart := vChart;
      vCandle.AxisX := 'x';
      vCandle.AxisY := 'y';
      vCandle.WickColor := $FF000000;
      vCandle.BodyWidthPercent := 60;
      vCandle.HollowUp := True;

      for i := 0 to 18 do
      begin
        vOpen := 100 + i;
        vClose := vOpen + (i mod 3) - 1;
```

```
        vCandle.AddOHLC(i, vOpen, vOpen + 3, vClose - 3, vClose);
      end;
      vCandle.UpdateLastOHLC(18, 118, 125, 110, 121);

      vOHLC.Chart := vChart;
      vOHLC.AxisX := 'x';
      vOHLC.AxisY := 'y';
      vOHLC.TickWidthPercent := 50;
      vOHLC.LineWidth := 2;
      for i := 0 to 18 do
        vOHLC.AddOHLC(i, 90 - i, 95 - i, 80 - i, 85 - i);

      Writeln('candlestick index: ', vCandle.Index);
      Writeln('ohlc index: ', vOHLC.Index);

      vRaster := TsgcUICanvas_Raster.Create(600, 300);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 600, 300));
        vRaster.SaveToPNGFile('candles.png');
      finally
        vRaster.Free;
      end;
    finally
      vCandle.Free;
      vOHLC.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChart
- TsgcChartSeries
- Panes, para dividir una serie de velas en su propio panel debajo del eje de precios

TsgcChartSeries_Pie

Unidad: sgcCharts_Series_Pie
TsgcChartSeries_Pie dibuja un gráfico circular o, con **InnerRadiusPercent** mayor que 0, un gráfico de anillo. Desciende de **TsgcChartSeries_Category** (unidad `sgcCharts_Series_Category`), la base de toda serie que contiene valores con nombre: un elemento por sector, guardado en la colección **Slices**, que es también el editor en tiempo de diseño y un almacén seguro entre hilos para datos en vivo. Un gráfico circular es una serie de celda: recibe su propia celda del área de trazado y no dibuja ejes, véase Áreas de trazado.

Propiedades

Propiedad	Tipo	Descripción
Slices	TsgcChart-CategoryI-tems	Los sectores, uno **TsgcChartPieSlice** por cada uno (**Text**, **Value**, **Color** y **Exploded**). Los colores de los sectores siguen el índice de la colección, de modo que un sector conserva su color con datos en vivo.
InnerRa-diusPer-cent	Double	0 (predeterminado) dibuja un gráfico circular; hasta 90 dibuja un anillo cuyo hueco es este porcentaje del radio.
StartAngle	Double	Ángulo del borde del primer sector en grados, 0 (predeterminado) son las 12 en punto.
Clockwise	Boolean	Valor predeterminado True.
Labels	TsgcChartS-liceLabels	`cslNone` , `cslInside` , `cslOutside` (predeterminado) o `cslAuto` . Véase Etiquetas más abajo.
LabelFor-mat	string	Valor predeterminado `{text} ({percent})` . Los tokens se escriben como un nombre entre llaves: `{text}` , `{value}` y `{percent}` .
ValueFor-mat	string	Máscara FormatFloat del token de valor; vacío (predeterminado) muestra hasta 2 decimales.
PercentDe-cimals	Integer	Decimales del token de porcentaje. Valor predeterminado 1.
LabelStyle	TsgcUITex-tStyle	Fuente de las etiquetas. **Color** 0 usa el color de texto del tema fuera del gráfi-co, y texto blanco u oscuro, elegido según el color del sector, dentro de él.
LeaderLi-neColor	TsgcUIColor	Color de las líneas guía de las etiquetas exteriores; 0 (predeterminado) usa el color del sector.
BorderCo-lor	TsgcUIColor	Color de los bordes de los sectores; 0 (predeterminado) usa el color de fondo del gráfico.
Border-Width	Double	Ancho de los bordes de los sectores. Valor predeterminado 1.
Explo-deOffset	Integer	Distancia a la que se separa un sector explotado, en píxeles de dispositivo a escala 1. Valor predeterminado 10.
Explo-deOnClick	Boolean	Valor predeterminado False. Con True, un clic en un sector alterna su indica-dor **Exploded** (**OthersExploded** para el sector Otros).

OthersThreshold	Double	Un porcentaje, 0 (predeterminado) está desactivado. Los sectores por debajo de él se fusionan en un único sector Otros, dibujado el último, cuando al menos dos cumplen la condición.
OthersText	string	Texto del sector Otros. Valor predeterminado `Others`.
OthersColor	TsgcUIColor	Color del sector Otros; 0 (predeterminado) usa el color de texto atenuado del tema.
OthersExploded	Boolean (public)	El sector Otros se separa del gráfico.
Total	Double (solo lectura)	Suma de los valores dibujados en el último renderizado.

Métodos

Método	Devuelve	Descripción
SliceColor(aIndex: Integer)	TsgcUIColor	El color con el que se dibuja el sector **aIndex** (un índice de la colección, o **CS_CHART_OTHERS_INDEX**): su **Color**, o si no el color de la paleta del gráfico para **aIndex**. El sector Otros usa **OthersColor**, o si no el color de texto atenuado del tema.

Métodos de las series de categorías

Las series de gráfico circular, radar, embudo y treemap comparten estos métodos de **TsgcChartSeries_Category**. Todos ellos se pueden llamar desde cualquier hilo: el almacén tiene su propio bloqueo, y una escritura que no cambia un valor no repinta.

Método	Devuelve	Descripción
SetValue(const aText: string; aValue: Double)	Integer	Establece el valor del primer elemento con este texto (distingue mayúsculas de minúsculas), o añade un elemento nuevo. Devuelve el índice del elemento. El último valor de un texto prevalece; los valores nunca se acumulan.
SetValueAt(aIndex: Integer; aValue: Double)		Establece el valor del elemento en **aIndex**, añadiendo elementos con texto vacío y valor 0 hasta **aIndex** cuando la colección es más corta.
AddItem(const aText: string; aValue: Double; aColor: TsgcUIColor = 0)	Integer	Añade un elemento, también cuando el texto ya está en uso. Devuelve su índice.
DeleteItem(aIndex: Integer)		Elimina un elemento.
ClearItems		Elimina todos los elementos.
IndexOfText(const aText: string)	Integer	Índice del primer elemento con este texto (distingue mayúsculas de minúsculas), -1 si no hay ninguno.
ItemCount	Integer	Número de elementos.

BeginUpdate / EndUpdate		Las escrituras entre ambos repintan el gráfico una sola vez, en el último **EndUpdate**.
BeginReload / EndReload		Marcar y barrer: el último **EndReload** elimina los elementos que no se escribieron desde el primer **BeginReload**, de modo que una recarga conserva los colores y los indicadores de los elementos que permanecen. También es una actualización, con un solo repintado al final.
GetSnapshot(var aSnap: TsgcChartCategorySnapshot)		Copia los elementos en **aSnap** bajo el bloqueo cuando han cambiado desde la última copia.

Los tokens de etiqueta de toda serie de categorías los expande **sgcChartFormatLabel**, que recibe el porcentaje y la proporción previa en unidades de porcentaje (25 significa 25%) y añade el signo `%`.

Etiquetas

cslInside centra cada etiqueta en su sector. **cslOutside** coloca las etiquetas en dos columnas, a izquierda y derecha del gráfico, unidas a sus sectores por líneas guía; las etiquetas de una columna se reparten para que nunca se solapen. **cslAuto** dibuja una etiqueta dentro cuando cabe en la cuerda y en el grosor del anillo, y fuera en caso contrario. Una etiqueta más ancha que su espacio se acorta para que quepa.

HitTest, información sobre herramientas y leyenda

Un impacto en un sector rellena **Index** y **SubIndex** con el índice de la colección (**CS_CHART_OTHERS_INDEX**, -2, para el sector Otros), **YValue** con el valor, **Percent** con la proporción y **Text** con el texto del sector. La información sobre herramientas muestra `{text}: {value} ({percent})`. La leyenda muestra una entrada por cada sector dibujado, con Otros al final. Véase TsgcChartInteraction.

Ejemplo: un anillo con Otros y un sector explotado

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_Series_Category, sgcCharts_Series_Pie;

var
  vChart: TsgcChart;
  vDonut: TsgcChartSeries_Pie;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vDonut := TsgcChartSeries_Pie.Create(nil);
    try
      vDonut.Chart := vChart;
      vDonut.Title := 'Revenue by region';
      vDonut.InnerRadiusPercent := 55; // a donut ring
      vDonut.Labels := cslAuto;
      vDonut.LabelFormat := '{text}: {value} ({percent})';
      vDonut.ValueFormat := '#,##0';
      vDonut.OthersThreshold := 5; // slices below 5% go into Others
      vDonut.ExplodeOnClick := True;

      // one repaint for the whole batch
      vDonut.BeginUpdate;
      try
        vDonut.SetValue('North', 420);
        vDonut.SetValue('South', 310);
        vDonut.SetValue('East', 275);
```

```
        vDonut.SetValue('West', 190);
        vDonut.SetValue('Islands', 12);
        vDonut.SetValue('Export', 9);
      finally
        vDonut.EndUpdate;
      end;
      // a live update: the latest value of a text wins
      vDonut.SetValue('South', 335);
      // pull the first slice out of the ring
      TsgcChartPieSlice(vDonut.Slices[0]).Exploded := True;

      vRaster := TsgcUICanvas_Raster.Create(640, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 640, 400));
        vRaster.SaveToPNGFile('donut.png');
      finally
        vRaster.Free;
      end;

      Writeln('total: ', FormatFloat('0', vDonut.Total));
      for i := 0 to vDonut.ItemCount - 1 do
        Writeln(vDonut.Slices[i].Text, ' $', IntToHex(vDonut.SliceColor(i), 8));
    finally
      vDonut.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChartSeries
- Áreas de trazado
- TsgcChartInteraction
- Título, Leyenda y Márgenes

TsgcChartSeries_Radar

Unidad: sgcCharts_Series_Radar
TsgcChartSeries_Radar dibuja valores sobre radios alrededor de un centro, como una línea poligonal o un polígono relleno. Es una serie de categorías (véanse los métodos de las series de categorías en el tema del gráfico circular): cada elemento de **Points** es un radio. Un radar da a su área de trazado el modo polar, y todos los radares de un área comparten una misma cuadrícula polar cuyos radios son la unión de los textos de sus elementos, por orden de primer uso. La escala radial es el **YAxis** de la serie, por lo que se aplican **Automatic**, **Minimum**, **Maximum**, el número de marcas y el formato de etiqueta de ese eje; los valores fuera de su rango se limitan al borde.

Propiedades

Propiedad	Tipo	Descripción
Points	TsgcChartCategoryItems	Un elemento por radio: **Text** da nombre al radio y **Value** es el valor radial.
Filled	Boolean	Valor predeterminado False. True rellena el polígono.
FillOpacity	Byte	Alfa del relleno. Valor predeterminado $40.
LineWidth	Double	Ancho del contorno en píxeles de dispositivo a escala 1; 0 no dibuja contorno.
Markers	TsgcChartSeriesMarkers	Marcadores de los vértices, visibles de forma predeterminada.
GridShape	TsgcChartRadarGrid	Anillos `crgPolygon` (predeterminado) o `crgCircle`. Se usa cuando este es el primer radar dibujado de su área.
StartAngle	Double	Ángulo del primer radio en grados, 0 (predeterminado) son las 12 en punto, en sentido horario. Se usa cuando este es el primer radar dibujado de su área.
SpokeLabelStyle	TsgcUITextStyle	Fuente de los textos de los radios; **Color** 0 usa el color de texto del tema. Se usa cuando este es el primer radar dibujado de su área.

Las propiedades **GridVisible** y **GridColor** del eje Y controlan los anillos y los radios, y su propiedad **Visible** activa o desactiva las etiquetas de las marcas. Se añade un anillo exterior cuando ninguna marca cae en el borde.

Geometría del último renderizado

Método	Devuelve	Descripción
CenterX, CenterY, Radius	Double (solo lectura)	Centro y radio de la cuadrícula polar.
SpokeCount	Integer (solo lectura)	Número de radios.
SpokeText[Index]	string (solo lectura)	Texto de un radio.

VertexPoint(aSpoke: Integer; out aX, aY: Double)	Boolean	Píxel del vértice de esta serie en el radio **aSpoke**. False (y el centro) para un radio inexistente, un valor que no es finito, o cuando la serie no se dibujó.

HitTest e información sobre herramientas

Un impacto devuelve el vértice más cercano de los radares del área dentro del radio de clic, a partir de la geometría del último renderizado. **Index** es el radio, **SubIndex** el elemento, **YValue** el valor, **Text** el texto del radio y **Distance** la distancia en píxeles; **Percent** y **ZValue** son NaN. La información sobre herramientas muestra `Product A, Speed: 80`.

Ejemplo: dos productos en seis criterios

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_Axis, sgcCharts_Series_Category,
  sgcCharts_Series_Radar;

const
  CS_CRITERIA: array [0 .. 5] of string = ('Speed', 'Price', 'Support',
    'Security', 'Features', 'Design');
  CS_PRODUCT_A: array [0 .. 5] of Double = (80, 65, 90, 70, 85, 60);
  CS_PRODUCT_B: array [0 .. 5] of Double = (60, 90, 70, 85, 55, 75);

var
  vChart: TsgcChart;
  vAxis: TsgcChartAxis;
  vA, vB: TsgcChartSeries_Radar;
  vRaster: TsgcUICanvas_Raster;
  vX, vY: Double;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    // the radial scale of a radar is its Y axis
    vAxis := vChart.Axes.Find('Y');
    vAxis.Automatic := False;
    vAxis.Minimum := 0;
    vAxis.Maximum := 100;

    vA := TsgcChartSeries_Radar.Create(nil);
    vB := TsgcChartSeries_Radar.Create(nil);
    try
      vA.Chart := vChart;
      vA.Title := 'Product A';
      vA.Filled := True;
      vA.FillOpacity := $50;
      // the first radar of the area decides the grid
      vA.GridShape := crgCircle;

      vB.Chart := vChart;
      vB.Title := 'Product B';
      vB.LineWidth := 2;

      for i := 0 to High(CS_CRITERIA) do
      begin
        vA.SetValue(CS_CRITERIA[i], CS_PRODUCT_A[i]);
        vB.SetValue(CS_CRITERIA[i], CS_PRODUCT_B[i]);
      end;

      vRaster := TsgcUICanvas_Raster.Create(500, 500);
      try
```

```
        vChart.Render(vRaster, sgcUIRect(0, 0, 500, 500));
        vRaster.SaveToPNGFile('radar.png');
      finally
        vRaster.Free;
      end;

      // the geometry of the last render
      Writeln('spokes: ', vA.SpokeCount, ', radius: ',
        FormatFloat('0', vA.Radius));
      for i := 0 to vA.SpokeCount - 1 do
        if vA.VertexPoint(i, vX, vY) then
          Writeln(vA.SpokeText[i], ' at ', FormatFloat('0', vX), ', ',
            FormatFloat('0', vY));
    finally
      vB.Free;
      vA.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChartSeries
- Áreas de trazado
- TsgcChartInteraction
- Ejes

TsgcChartSeries_Heatmap

Unidad: sgcCharts_Series_Heatmap
TsgcChartSeries_Heatmap dibuja una matriz de valores como celdas coloreadas sobre los ejes X e Y. Es una serie XY (descendiente directa de **TsgcChartSeries**), por lo que comparte su trazado con series de línea, barras o dispersión y usa ejes numéricos, de fecha y hora o de categorías tal como son. Los valores residen en un almacén seguro entre hilos de líneas, **Data** (**TsgcChartHeatmapData**): con **Orientation** `chmColumns` una línea es una columna en una clave X y sus celdas se extienden a lo largo de Y; con `chmRows` una línea es una fila en una clave Y. Con **Capacity** mayor que 0 el almacén conserva solo las líneas más recientes, la idea del búfer circular aplicada a una matriz, que convierte un mapa de calor en un espectrograma continuo. Los colores proceden de una escala de color.

Propiedades

Propiedad	Tipo	Descripción
Orientation	TsgcChartHeatmapOrientation	`chmColumns` (predeterminado) o `chmRows`.
Capacity	Integer	0 (predeterminado) deja crecer el almacén; mayor que 0 conserva solo las líneas más recientes.
CellOrigin	Double	Centro de la celda 0 en el eje de celdas. Valor predeterminado 0.
CellStep	Double	Distancia entre los centros de las celdas, mayor que 0. Valor predeterminado 1.
LineStep	Double	Extensión de una línea en el eje de líneas; 0 (predeterminado) llega hasta la mitad de la distancia a las líneas vecinas.
ColorScale	TsgcChartColorScale	Colores de las celdas. Un **MaxColor** de 0 se resuelve en el **Color** de la serie, o si no en el primer color de la paleta; un **MinColor** de 0 en el fondo del trazado. Véase Escala de color.
ShowValues	Boolean	Valor predeterminado False. True escribe el valor en cada celda pintada de la ruta vectorial.
ValueFormat	string	Máscara FormatFloat de los textos de valor; vacío (predeterminado) muestra hasta 2 decimales.
ValueStyle	TsgcUITextStyle	Fuente de los textos de valor; **Color** 0 elige texto blanco u oscuro según la luminancia de la celda.
CellBorderColor	TsgcUIColor	Borde dibujado dentro de cada celda pintada de la ruta vectorial; 0 (predeterminado) no dibuja ninguno.
Data	TsgcChartHeatmapData (solo lectura)	El almacén de líneas. Las escrituras hechas directamente en **Data** no repintan el gráfico; los métodos de la serie que se indican a continuación sí.
Cells[aLine, aCell]	Double	El valor de una celda; NaN fuera de rango. Una escritura fuera de rango se ignora. NaN es una celda nula, que nunca se pinta ni recibe impactos.
ImagePath	Boolean (solo lectura)	True cuando el último renderizado rellenó una imagen, False cuando dibujó rectángulos.

DrawnCells	Integer (solo lectura)	Celdas visibles del último renderizado.

Métodos

Método	Devuelve	Descripción
BeginUpdate / EndUpdate		Las escrituras entre ambos repintan el gráfico una sola vez. Cualquier hilo.
Clear		Elimina todas las líneas.
SetSize(aLines, aCells: Integer)		Dimensiona la matriz a **aLines** líneas de **aCells** celdas.
AppendLine(const aValues: array of Double)		Añade una línea con la clave del contador de líneas en curso del almacén.
AppendLine(const aKey: Double; const aValues: array of Double)		Añade una línea en la clave X (columnas) o Y (filas) **aKey**, por ejemplo una marca de tiempo. Una línea más larga que el número actual de celdas ensancha todas las líneas y las rellena con celdas nulas.
SetLineLabels(const aLabels: array of string) / SetLineLabels(aLabels: TStrings)		Textos de las líneas, mostrados por un eje de líneas de categorías.
SetCellLabels(const aLabels: array of string) / SetCellLabels(aLabels: TStrings)		Textos de las celdas, mostrados por un eje de celdas de categorías.

Ejes y dibujo

En un eje de categorías una línea o una celda se sitúa en su índice lógico, sea cual sea su clave o **CellOrigin**, de modo que un almacén continuo permanece alineado con sus etiquetas, que llegan al eje a través de **GetDataCategories**. Cada borde de celda se ajusta a un píxel de dispositivo entero. La serie elige la ruta de dibujo según el número y el tamaño de las celdas visibles, nunca según el lienzo: hasta **CS_CHART_HEATMAP_VECTOR_CELLS** (20.000) celdas de al menos **CS_CHART_HEATMAP_VECTOR_MIN_SIZE** (3) píxeles se dibujan como rectángulos con bordes y textos de valor; las matrices mayores se rellenan como una sola imagen, sin bordes ni textos. El resultado es el mismo en los lienzos raster, SVG, VCL y FMX.

HitTest, información sobre herramientas y leyenda

Un impacto rellena **Area** con `chaPlot`, **Index** con la línea, **SubIndex** con la celda, **XValue** e **YValue** con el centro de la celda, **ZValue** con el valor y **Text** con `línea, celda` (el texto del eje cuando una etiqueta está vacía); **Distance** es 0 y **Percent** es NaN. La información sobre herramientas muestra `Sales, Mar, South: 64`. Cuando se dibuja una serie de líneas encima, un punto de línea dentro del radio de clic prevalece sobre la celda. La leyenda muestra una barra de color con el mínimo y el máximo de la escala.

Ejemplo: una matriz de región por mes

```
uses
  SysUtils, Math,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Types, sgcCharts_Chart, sgcCharts_HeatmapData,
  sgcCharts_Series_Heatmap;
```

```
var
  vChart: TsgcChart;
  vHeat: TsgcChartSeries_Heatmap;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    // category axes: one line per month on X, one cell per region on Y
    vChart.Axes.Find('X').AxisType := catCategory;
    vChart.Axes.Find('Y').AxisType := catCategory;

    vHeat := TsgcChartSeries_Heatmap.Create(nil);
    try
      vHeat.Chart := vChart;
      vHeat.Title := 'Sales';
      vHeat.Orientation := chmColumns;
      vHeat.ColorScale.MinColor := $FFE3F2FD;
      vHeat.ColorScale.MaxColor := $FF0D47A1;
      vHeat.ShowValues := True;
      vHeat.ValueFormat := '0';
      vHeat.CellBorderColor := $FFFFFFFF;

      // one repaint for the whole matrix
      vHeat.BeginUpdate;
      try
        vHeat.SetCellLabels(['North', 'South', 'East', 'West']);
        vHeat.SetLineLabels(['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']);
        for i := 0 to 5 do
          vHeat.AppendLine([40 + i * 5, 30 + i * 3, 55 - i * 4, 20 + i * 8]);
      finally
        vHeat.EndUpdate;
      end;
      // a null cell is not painted and never hit
      vHeat.Cells[2, 1] := NaN;

      vRaster := TsgcUICanvas_Raster.Create(700, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 700, 400));
        vRaster.SaveToPNGFile('heatmap.png');
      finally
        vRaster.Free;
      end;

      Writeln('cells drawn: ', vHeat.DrawnCells, ', one image: ',
        vHeat.ImagePath);
      Writeln('Apr, West: ', FormatFloat('0', vHeat.Cells[3, 3]));
    finally
      vHeat.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChartSeries
- Áreas de trazado
- TsgcChartInteraction
- Escala de color
- Ejes

TsgcChartSeries_Treemap

Unidad: sgcCharts_Series_Treemap
TsgcChartSeries_Treemap dibuja una jerarquía como rectángulos anidados cuyas áreas siguen los valores. Es una serie de categorías (véanse los métodos de las series de categorías en el tema del gráfico circular): el **Text** de cada elemento de **Nodes** es una ruta como `Europe/Spain/Madrid`, dividida por **PathDelimiter**, y los grupos son las sumas de sus hijos. La disposición se calcula por niveles con el algoritmo squarified, con relleno y franjas de encabezado, y conserva el orden de los hermanos durante las actualizaciones de valores en vivo. Un clic profundiza en un grupo y una ruta de navegación sobre los mosaicos permite volver. Un treemap es una serie de celda, véase Áreas de trazado.

Propiedades

Propiedad	Tipo	Descripción
Nodes	TsgcChartCategoryItems	Las hojas: una ruta en **Text**, el valor en **Value**. El **Color** de un elemento de nivel superior sustituye a su color de la paleta.
PathDelimiter	Char	Separador de los segmentos de la ruta. Valor predeterminado `/`.
MaxDepth	Integer	Niveles dibujados por debajo de la raíz actual; menor que 1 significa sin límite. Valor predeterminado 2. Los grupos más profundos se dibujan plegados, como un solo mosaico.
ColorMode	TsgcChartTreemapColorMode	`ctcGroup` (predeterminado): un color de la paleta por grupo de nivel superior, aclarado (tema claro) u oscurecido (tema oscuro) según el nivel. `ctcValue`: mosaicos coloreados mediante **ColorScale**, sobre el rango de los mosaicos situados bajo la raíz.
ColorScale	TsgcChartColorScale	Colores de `ctcValue`; un **MaxColor** de 0 es el **Color** de la serie, o si no el primer color de la paleta. Véase Escala de color.
Levels	TsgcChartTreemapLevels	Borde, relleno y encabezado por nivel; **Items[0]** es el nivel 1, los hijos de la raíz actual. Véase TsgcChartTreemapLevel más abajo.
LabelFormat	string	Valor predeterminado `{text}`. Los tokens se escriben como un nombre entre llaves: `{text}`, `{value}` y `{percent}`; text es el último segmento de la ruta y percent es la proporción de la raíz actual.
ValueFormat	string	Máscara FormatFloat del token de valor; vacío (predeterminado) muestra hasta 2 decimales.
PercentDecimals	Integer	Valor predeterminado 1.
LabelStyle	TsgcUITextStyle	Fuente de las etiquetas; **Color** 0 elige texto blanco u oscuro según el relleno.
StableLayout	Boolean	Valor predeterminado True: una actualización de valor conserva el orden de los hermanos, de modo que los mosaicos no saltan con datos en vivo.
DrillOnClick	Boolean	Valor predeterminado True: un clic en un mosaico profundiza en su ancestro de nivel 1, un clic en una miga de pan sube hasta ella.

OnDrill	TNotifyEvent	Se dispara cuando cambia **DrillPath**.
DrillPath	string (public)	Ruta de la raíz actual; vacío es la raíz del árbol. Al establecerla se dispara **OnDrill**.
Layout	TsgcChartTreemapLayout (solo lectura)	El árbol y la geometría del último renderizado: nodos, nodos dibujados, la ruta de navegación (**Crumbs**) y su cuadro.

TsgcChartTreemapLevel

Propiedad	Tipo	Descripción
BorderColor	TsgcUIColor	0 (predeterminado) usa el color de fondo del gráfico.
BorderWidth	Double	Píxeles de dispositivo a escala 1; 0 no dibuja borde.
Padding	Double	Espacio entre un grupo de este nivel y sus hijos, en píxeles de dispositivo a escala 1.
ShowHeader	Boolean	Una franja de encabezado con el nombre del grupo sobre los hijos.

Un nuevo elemento de nivel toma los valores predeterminados de su nivel: 2 píxeles de relleno y un encabezado para el primer elemento, 1 píxel y sin encabezado para los demás. Un nivel sin elemento usa un borde de 1 píxel.

Métodos

Método	Devuelve	Descripción
DrillDown(const aPath: string)		Convierte la ruta de un nodo en la raíz actual; una ruta vacía es la raíz del árbol.
DrillUp		Convierte el padre de la raíz actual en la raíz actual.
DrillReset		Convierte la raíz del árbol en la raíz actual.
NodeColor(aNode: Integer)	TsgcUIColor	Relleno del nodo de disposición **aNode** en el último renderizado, 0 cuando no se dibuja.
CrumbRect(aIndex: Integer)	TsgcUIRect	Cuadro de la miga de pan **aIndex** (en el orden de **Layout.Crumbs**) en el último renderizado; vacío cuando no se dibuja. La primera miga es la raíz, con el texto `All` .

Etiquetas, HitTest y leyenda

Una etiqueta se muestra cuando su cuadro es al menos tan ancho como tres caracteres y tan alto como una línea; un texto más largo se acorta para que quepa, y un mosaico más pequeño no muestra etiqueta. Un impacto devuelve el mosaico más profundo bajo el punto: **Index** es el elemento (-1 para un grupo, que no tiene elemento propio), **SubIndex** la profundidad, **YValue** el valor, **Percent** la proporción de la raíz actual y **Text** la ruta. Un impacto en una miga de pan tiene **Percent** NaN. En `ctcGroup` la leyenda enumera los grupos de nivel superior de todo el árbol; en `ctcValue` muestra una barra de color.

Ejemplo: un árbol de tres niveles con navegación en profundidad

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_Series_Category, sgcCharts_Series_Treemap;

var
  vChart: TsgcChart;
  vTree: TsgcChartSeries_Treemap;
  vLevel: TsgcChartTreemapLevel;
  vRaster: TsgcUICanvas_Raster;
begin
  vChart := TsgcChart.Create(nil);
  try
    vTree := TsgcChartSeries_Treemap.Create(nil);
    try
      vTree.Chart := vChart;
      vTree.Title := 'Sales';
      vTree.PathDelimiter := '/';
      vTree.MaxDepth := 2; // continents and countries below the root
      vTree.ColorMode := ctcGroup;
      vTree.LabelFormat := '{text} {percent}';
      // level 1: 3 pixels of padding and a header strip
      vLevel := vTree.Levels.Add;
      vLevel.Padding := 3;
      vLevel.ShowHeader := True;

      vTree.BeginUpdate;
      try
        vTree.SetValue('Europe/Spain/Madrid', 120);
        vTree.SetValue('Europe/Spain/Barcelona', 95);
        vTree.SetValue('Europe/France/Paris', 140);
        vTree.SetValue('Europe/France/Lyon', 45);
        vTree.SetValue('America/USA/New York', 160);
        vTree.SetValue('America/USA/Chicago', 70);
        vTree.SetValue('America/Mexico/Monterrey', 40);
        vTree.SetValue('Asia/Japan/Tokyo', 130);
        vTree.SetValue('Asia/India/Mumbai', 60);
      finally
        vTree.EndUpdate;
      end;

      vRaster := TsgcUICanvas_Raster.Create(800, 500);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 500));
        vRaster.SaveToPNGFile('treemap.png');
        // drill into one continent: a breadcrumb shows the path
        vTree.DrillDown('Europe');
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 500));
        vRaster.SaveToPNGFile('treemap_europe.png');
      finally
        vRaster.Free;
      end;

      Writeln('root: ', vTree.DrillPath);
      vTree.DrillUp;
      Writeln('root after DrillUp: "', vTree.DrillPath, '"');
    finally
      vTree.Free;
    end;
  finally
    vChart.Free;
```

```
  end;
end.
```

Véase también

- TsgcChartSeries
- Áreas de trazado
- TsgcChartInteraction
- Escala de color

TsgcChartSeries_Funnel

Unidad: sgcCharts_Series_Funnel
TsgcChartSeries_Funnel dibuja las etapas de un proceso, como un embudo de ventas, como un embudo que se estrecha hasta un cuello. Es una serie de categorías (véanse los métodos de las series de categorías en el tema del gráfico circular): cada elemento de **Stages** es una etapa, de arriba abajo en el orden de la colección. Un embudo es una serie de celda, véase Áreas de trazado.

Propiedades

Propiedad	Tipo	Descripción
Stages	TsgcChartCategoryItems	Las etapas, de arriba abajo.
NeckWidthPercent	Double	Ancho del cuello, porcentaje del ancho del embudo. Valor predeterminado 30.
NeckHeightPercent	Double	Alto del cuello, porcentaje del alto del embudo. Valor predeterminado 25.
ProportionalHeight	Boolean	False (predeterminado): etapas de igual altura. True: alturas según el valor.
Gap	Integer	Espacio entre dos etapas, en píxeles de dispositivo a escala 1. Valor predeterminado 2.
Labels	TsgcChartSliceLabels	`cslOutside` (predeterminado) dibuja una columna a la derecha con líneas guía del color de la etapa; `cslInside`, `cslAuto` y `cslNone` funcionan como en el gráfico circular.
LabelFormat	string	Valor predeterminado `{text}: {value} ({percent})`. Los tokens se escriben como un nombre entre llaves: `{text}`, `{value}` y `{percent}` (proporción de la primera etapa) y `{previous}` (proporción de la etapa superior, 100% para la primera).
ValueFormat	string	Máscara FormatFloat del token de valor; vacío (predeterminado) muestra hasta 2 decimales.
PercentDecimals	Integer	Valor predeterminado 1.
LabelStyle	TsgcUITextStyle	**Color** 0 usa el color de texto del tema fuera del embudo, y texto blanco u oscuro, elegido según el color de la etapa, dentro de él.
BorderColor	TsgcUIColor	0 (predeterminado) usa el color de fondo del gráfico.
BorderWidth	Double	Valor predeterminado 1.

Métodos y geometría del último renderizado

Método	Devuelve	Descripción

StageColor(aIndex: Integer)	TsgcUIColor	El color con el que se dibuja la etapa **aIndex**: su **Color**, o si no el color de la paleta del gráfico para **aIndex**.
HalfWidthAt(aY: Double)	Double	Semiancho de la forma en **aY**; 0 por encima o por debajo del embudo.
StageAt(X, Y: Double)	Integer	Posición de la etapa dibujada (o de su etiqueta exterior) en **X, Y**, -1 si no hay ninguna.
DrawnCount	Integer (solo lectura)	Número de etapas dibujadas.
DrawnStages[aPos]	TsgcChartFunnelStage (solo lectura)	Una etapa dibujada, de arriba abajo: **Index** (índice de la colección), **Value**, **Percent** (de la primera etapa), **Previous** (de la etapa superior), **Top**, **Bottom**, los semianchos, el lugar y el cuadro de la etiqueta, y los puntos de la línea guía.
CenterX, FunnelRect, NeckTop, NeckWidth	Double, TsgcUIRect (solo lectura)	El centro horizontal, el rectángulo delimitador de la forma, la parte superior del cuello y su ancho.

HitTest, información sobre herramientas y leyenda

Un impacto en una etapa o en su etiqueta exterior rellena **Index** y **SubIndex** con el índice de la colección, **YValue** con el valor, **Percent** con la proporción de la primera etapa y **Text** con el texto de la etapa. La información sobre herramientas muestra `{text}: {value} ({percent})`. La leyenda muestra una entrada por cada etapa dibujada.

Ejemplo: un embudo de ventas

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_Series_Category, sgcCharts_Series_Funnel;

var
  vChart: TsgcChart;
  vFunnel: TsgcChartSeries_Funnel;
  vStage: TsgcChartFunnelStage;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vFunnel := TsgcChartSeries_Funnel.Create(nil);
    try
      vFunnel.Chart := vChart;
      vFunnel.Title := 'Sales pipeline';
      vFunnel.NeckWidthPercent := 25;
      vFunnel.NeckHeightPercent := 20;
      vFunnel.Labels := cslOutside;
      vFunnel.LabelFormat := '{text}: {value} ({previous} of the stage above)';
      vFunnel.ValueFormat := '#,##0';

      vFunnel.BeginUpdate;
      try
        vFunnel.SetValue('Visits', 12000);
        vFunnel.SetValue('Sign-ups', 4200);
        vFunnel.SetValue('Trials', 1800);
        vFunnel.SetValue('Quotes', 640);
        vFunnel.SetValue('Orders', 310);
      finally
        vFunnel.EndUpdate;
```

```
      end;

      vRaster := TsgcUICanvas_Raster.Create(700, 420);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 700, 420));
        vRaster.SaveToPNGFile('funnel.png');
      finally
        vRaster.Free;
      end;

      // the stages of the last render, top to bottom
      for i := 0 to vFunnel.DrawnCount - 1 do
      begin
        vStage := vFunnel.DrawnStages[i];
        Writeln(vFunnel.Stages[vStage.Index].Text, ': ',
          FormatFloat('0.0', vStage.Percent), '% of the first stage');
      end;
    finally
      vFunnel.Free;
    end;
  finally
    vChart.Free;
  end;
end.
```

Véase también

- TsgcChartSeries
- Áreas de trazado
- TsgcChartInteraction

TsgcVCLChart

Unidad: sgcCharts_VCL
TsgcVCLChart es la vista VCL de un modelo TsgcChart: un descendiente de `TCustomControl`, solo Windows, que pinta el gráfico asignado a su propiedad **Chart**. El control registra un oyente en el gráfico y se repinta cada vez que el modelo cambia; el propio modelo no contiene código de dibujo, así que el mismo gráfico también puede mostrarse con un TsgcFMXChart o renderizarse sin interfaz visual con un TsgcChartRenderer al mismo tiempo.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este control. Asignarlo registra el control como oyente; borrarlo, o destruir el gráfico, hace que el control deje de pintar nada.
FrameCount	Cardinal (solo lectura)	Número de fotogramas completos renderizados desde que se creó el control.
OverlayFrameCount	Cardinal (solo lectura)	Número de fotogramas solo de superposición renderizados desde que se creó el control. Véase **Repintados solo de superposición** a continuación.

Las propiedades habituales de `TCustomControl` también se publican sin cambios: **Align**, **Anchors**, **Color**, **Constraints**, **DragCursor**, **DragKind**, **DragMode**, **Enabled**, **ParentColor**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop**, **Visible**, y los eventos de ratón, arrastre y redimensionado.

Métodos

Método	Devuelve	Descripción
SaveToPNG(const aFileName: string)		Renderiza **Chart** al tamaño de cliente y PPI actuales del control en un nuevo archivo PNG. No hace nada cuando **Chart** es nil.
SaveToSVG(const aFileName: string)		Renderiza **Chart** al tamaño de cliente y PPI actuales del control en un archivo SVG. No hace nada cuando **Chart** es nil.

Limitación de frecuencia con MaxFPS

MaxFPS no es una propiedad de este control, pertenece al modelo de gráfico (véase TsgcChart). Una notificación de cambio del gráfico solo marca el control como sucio y programa un repintado, como máximo **Chart.MaxFPS** veces por segundo: cuando el siguiente fotograma ya toca, el control se invalida de inmediato; en caso contrario, un temporizador de un solo disparo lo invalida cuando el fotograma toque. Los fotogramas se ajustan a un plazo en curso (el plazo anterior más 1000 dividido por **MaxFPS**), no a un intervalo fijo desde el último pintado, de modo que la granularidad del temporizador de Windows no permite que la frecuencia efectiva caiga por debajo del límite. Una notificación que llega desde un hilo de alimentación nunca toca el control directamente, envía un mensaje privado que el hilo principal recoge antes de programar el repintado.

Repintados solo de superposición

El control también registra un segundo oyente, de superposición, en el gráfico. Un cambio que solo mueve la retícula, la información sobre herramientas o una selección de banda elástica (**TsgcChart.InvalidateOverlay**) no ejecuta un repintado completo: los ejes, las series, el fondo, el título y la leyenda no se redibujan. En su lugar, los píxeles del último render completo se copian de nuevo en el rasterizador interno y solo se dibuja encima **TsgcChart.RenderOverlay**, por eso **OverlayFrameCount** y **FrameCount** se contabilizan por separado. Un render completo sigue ocurriendo siempre que el control está sucio por otro motivo, como un redimensionado, un cambio de PPI, o un cambio en los datos, los ejes, el tema o el diseño del gráfico.

Ejemplo: representar un gráfico en un formulario VCL

```
uses
  Forms, Classes, SysUtils, Math, ExtCtrls, Controls,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcCharts_VCL;

type
  TfrmMain = class(TForm)
  private
    FChart: TsgcChart;
    FView: TsgcVCLChart;
    FSeries: TsgcChartSeries_Line;
  public
    constructor Create(aOwner: TComponent); override;
  end;

constructor TfrmMain.Create(aOwner: TComponent);
var
  i: Integer;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcCharts VCL Sample';
  Width := 820;
  Height := 480;

  FChart := TsgcChart.Create(Self);
  FChart.Axes.Add.Name := 'bottom';
  FChart.Axes[0].Position := capBottom;
  FChart.Axes.Add.Name := 'left';
  FChart.Axes[1].Position := capLeft;

  FSeries := TsgcChartSeries_Line.Create(Self);
  FSeries.Chart := FChart;
  FSeries.AxisX := 'bottom';
  FSeries.AxisY := 'left';
  for i := 0 to 99 do
    FSeries.AddXY(i, Sin(i / 10) * 50 + 50);

  FView := TsgcVCLChart.Create(Self);
  FView.Parent := Self;
  FView.Align := alClient;
  FView.Chart := FChart;
end;

// elsewhere, e.g. from a menu action or before closing:
FView.SaveToPNG('chart.png');
```

Véase también

- TsgcChart
- TsgcFMXChart
- TsgcChartRenderer

- TsgcChartSeries_Line
- TsgcChartInteraction
- Panes

TsgcFMXChart

Unidad: sgcCharts_FMX
TsgcFMXChart es la vista FireMonkey de un modelo TsgcChart: un descendiente de `TControl` que pinta el gráfico asignado a su propiedad **Chart** en Windows, macOS, iOS y Android. La unidad compila a nada a menos que se defina `SGC_CHARTS_FMX`, lo que requiere Delphi XE6 o posterior. El modelo no contiene código de dibujo, así que el mismo gráfico también puede mostrarse con un TsgcVCLChart o renderizarse sin interfaz visual con un TsgcChartRenderer al mismo tiempo.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este control. Asignarlo registra el control como oyente; borrarlo, o destruir el gráfico, hace que el control deje de pintar nada.
FrameCount	Cardinal (solo lectura)	Número de fotogramas completos renderizados desde que se creó el control.
OverlayFrameCount	Cardinal (solo lectura)	Número de fotogramas solo de superposición renderizados desde que se creó el control. Véase **Repintados solo de superposición** a continuación.

Las propiedades habituales de `TControl` también se publican sin cambios: **Align**, **Anchors**, **ClipChildren**, **ClipParent**, **Cursor**, **DragMode**, **EnableDragHighlight**, **Enabled**, **Locked**, **Height**, **HitTest**, **Padding**, **Opacity**, **Margins**, **PopupMenu**, **Position**, **RotationAngle**, **RotationCenter**, **Scale**, **Visible**, **Width**, **TouchTargetExpansion**, y los eventos de ratón, arrastre y redimensionado.

Métodos

Método	Devuelve	Descripción
SaveToPNG(const aFileName: string)		Renderiza **Chart** al tamaño y escala de escena actuales del control en un nuevo archivo PNG. No hace nada cuando **Chart** es nil.
SaveToSVG(const aFileName: string)		Renderiza **Chart** al tamaño y escala de escena actuales del control en un archivo SVG. No hace nada cuando **Chart** es nil.
DetectPixelOrder(var aData: TBitmapData) [class function]	TsgcFMXChartPixelOrder	Escribe un píxel de sonda en los datos de un mapa de bits mapeado y lo relee para averiguar si el mapa de bits de la plataforma almacena `fpoBGRA`, `fpoRGBA` u otro orden de canal (`fpoOther`). Se usa internamente antes de copiar los píxeles renderizados en un mapa de bits FMX; la mayoría de las aplicaciones nunca lo llaman directamente.
CopyToBitmapData(aRasterizer: TsgcUIRasterizer; var aData: TBitmapData) [class procedure]		Convierte los píxeles ARGB directos de **aRasterizer** en los píxeles premultiplicados de los datos de un mapa de bits mapeado, sobre el menor de los dos tamaños. Se usa internamente en **Paint**; la mayoría de las aplicaciones nunca lo llaman directamente.

Limitación de frecuencia con MaxFPS

MaxFPS no es una propiedad de este control, pertenece al modelo de gráfico (véase TsgcChart). Una notificación de cambio del gráfico solo marca el control como sucio y programa un repintado, como máximo **Chart.MaxFPS** veces por segundo: cuando el siguiente fotograma ya toca, el control se repinta de inmediato; en caso contrario, un temporizador de un solo disparo lo repinta cuando el fotograma toque. Los fotogramas se ajustan a un plazo en curso (el plazo anterior más 1000 dividido por **MaxFPS**), no a un intervalo fijo desde el último pintado. Una notificación que llega desde un hilo de alimentación nunca toca el control directamente, encola una llamada al hilo principal con **TThread.Queue** (como máximo una llamada pendiente), y el hilo principal programa el repintado; el destructor elimina una llamada que aún esté en cola.

Repintados solo de superposición

El control también registra un segundo oyente, de superposición, en el gráfico. Un cambio que solo mueve la retícula, la información sobre herramientas o una selección de banda elástica (**TsgcChart.InvalidateOverlay**) no ejecuta un repintado completo: los ejes, las series, el fondo, el título y la leyenda no se redibujan. En su lugar, los píxeles del último render completo se copian de nuevo en el rasterizador interno y solo se dibuja encima **TsgcChart.RenderOverlay**, por eso **OverlayFrameCount** y **FrameCount** se contabilizan por separado. Un render completo sigue ocurriendo siempre que el control está sucio por otro motivo, como un redimensionado, un cambio de escala de escena, o un cambio en los datos, los ejes, el tema o el diseño del gráfico.

Ejemplo: representar un gráfico en un formulario FMX

```
uses
  System.SysUtils, System.Types, System.Math, System.Classes,
  FMX.Types, FMX.Forms, FMX.Controls,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcCharts_FMX;

type
  TfrmMain = class(TForm)
  private
    FChart: TsgcChart;
    FView: TsgcFMXChart;
    FSeries: TsgcChartSeries_Line;
  public
    constructor Create(aOwner: TComponent); override;
  end;

constructor TfrmMain.Create(aOwner: TComponent);
var
  i: Integer;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcCharts FMX Sample';
  Width := 820;
  Height := 480;

  FChart := TsgcChart.Create(Self);
  FChart.Axes.Add.Name := 'bottom';
  FChart.Axes[0].Position := capBottom;
  FChart.Axes.Add.Name := 'left';
  FChart.Axes[1].Position := capLeft;

  FSeries := TsgcChartSeries_Line.Create(Self);
  FSeries.Chart := FChart;
  FSeries.AxisX := 'bottom';
  FSeries.AxisY := 'left';
  for i := 0 to 99 do
    FSeries.AddXY(i, Sin(i / 10) * 50 + 50);

  FView := TsgcFMXChart.Create(Self);
  FView.Parent := Self;
```

```
  FView.Align := TAlignLayout.Client;
  FView.Chart := FChart;
end;

// elsewhere, e.g. from an action or before closing:
FView.SaveToPNG('chart.png');
```

Véase también

- TsgcChart
- TsgcVCLChart
- TsgcChartRenderer
- TsgcChartSeries_Line
- TsgcChartInteraction
- Panes

TsgcChartRenderer

Unidad: sgcCharts_Renderer
TsgcChartRenderer es la vista sin interfaz visual de un modelo TsgcChart: un `TComponent` simple sin dependencia de VCL o FMX y sin ventana propia, de modo que renderiza un gráfico en un archivo o flujo PNG, un documento SVG o un búfer de píxeles ARGB en bruto desde un programa de consola, un servicio de Windows, un módulo ISAPI o un daemon Linux64. Al no tener superficie visual, no se ve afectado por **Chart.MaxFPS**: no hay bucle de fotogramas ni temporizador, cada llamada de render pinta el gráfico de forma síncrona en el momento en que se invoca.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este renderizador. Cada método Render* genera **EsgcChartRendererError** cuando esto es nil.
Width	Integer	Ancho lógico predeterminado en píxeles usado por las sobrecargas que no reciben tamaño. Valor predeterminado 800.
Height	Integer	Alto lógico predeterminado en píxeles usado por las sobrecargas que no reciben tamaño. Valor predeterminado 500.
Scale	Double	Píxeles de dispositivo por píxel lógico. Valor predeterminado 1. La salida es `Round(Width * Scale)` por `Round(Height * Scale)` píxeles de dispositivo; **Chart.Scale** se establece en este valor durante el render y se restaura a su valor anterior al finalizar.
CompressPNG	Boolean	Indica si la salida PNG se comprime con deflate. Valor predeterminado True cuando la biblioteca se compiló con `SGC_CHARTS_DEFLATE`, False en caso contrario.

Métodos

Método	Devuelve	Descripción
RenderToPNGFile(const aFileName: string) RenderToPNGFile(const aFileName: string; aWidth, aHeight: Integer)		Renderiza a un archivo PNG, con **Width**/**Height** o con el tamaño lógico indicado sin modificar **Width** y **Height**.
RenderToPNGStream(aStream: TStream) RenderToPNGStream(aStream: TStream; aWidth, aHeight: Integer)		Renderiza PNG (RGBA, 8 bits) a un flujo.
RenderToSVG RenderToSVG(aWidth, aHeight: Integer)	string	Renderiza y devuelve un documento SVG 1.1 (ASCII puro) como cadena.
RenderToSVGFile(const aFileName: string) RenderToSVGFile(const aFileName: string; aWidth, aHeight: Integer)		Renderiza SVG a un archivo.

RenderToSVGStream(aStream: TStream) RenderToSVGStream(aStream: TStream; aWidth, aHeight: Integer)		Renderiza SVG a un flujo.
RenderToBuffer(var aBuffer: TsgcUICardinalArray; out aPixelWidth, aPixelHeight: Integer) RenderToBuffer(var aBuffer: TsgcUICardinalArray; out aPixelWidth, aPixelHeight: Integer; aWidth, aHeight: Integer)		Renderiza en píxeles $AARRGGBB directos (no premultiplicados), de arriba abajo, con un stride de fila igual a **aPixelWidth**; **aPixelWidth** y **aPixelHeight** se devuelven con el tamaño de dispositivo.

Concurrencia

Todo método de render puede llamarse desde cualquier hilo. Cada **TsgcChartRenderer** posee sus propios lienzos, uno rasterizado y uno SVG, protegidos por un bloqueo privado de esa instancia del renderizador. Además, todo render realizado por cualquier **TsgcChartRenderer** del proceso pasa por otro bloqueo compartido por toda la unidad, porque **TsgcChart.Render** mantiene el estado de diseño en el propio gráfico mientras se ejecuta. Mientras un render está en curso, el gráfico no debe modificarse estructuralmente, es decir, no se deben añadir, quitar ni liberar series o ejes, ni cambiar ninguna propiedad que afecte al diseño; añadir datos a través de los búferes de las series (**Buffer.Append**, **OHLCBuffer.Append**, las fuentes de datos) es seguro, porque las series dibujan a partir de una instantánea tomada bajo el propio bloqueo del búfer.
El bloqueo de render compartido solo lo toma **TsgcChartRenderer**. Una vista visual, TsgcVCLChart o TsgcFMXChart, no lo toma, así que no se debe renderizar el mismo gráfico con un renderizador en un hilo mientras una vista visual lo está pintando en el mismo instante.

Ejemplo: render sin interfaz visual a PNG y SVG

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcCharts_Renderer;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  vRenderer: TsgcChartRenderer;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Title.Text := 'Headless Render';
    vChart.Axes.Add.Name := 'bottom';
    vChart.Axes[0].Position := capBottom;
    vChart.Axes.Add.Name := 'left';
    vChart.Axes[1].Position := capLeft;

    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Chart := vChart;
      vSeries.AxisX := 'bottom';
      vSeries.AxisY := 'left';
      for i := 0 to 99 do
        vSeries.AddXY(i, Sin(i / 10) * 100);

      vRenderer := TsgcChartRenderer.Create(nil);
      try
        vRenderer.Chart := vChart;
        vRenderer.Width := 800;
        vRenderer.Height := 400;
        vRenderer.RenderToPNGFile('renderer.png');
        vRenderer.RenderToSVGFile('renderer.svg');
      finally
        vRenderer.Free;
```

```
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase también

- TsgcChart
- TsgcVCLChart
- TsgcFMXChart
- TsgcChartSeries_Line
- Panes

TsgcChartInteraction

Unidad: sgcCharts_Interaction
TsgcChartInteraction es la API de entrada de un TsgcChart: un `TPersistent`, sin dependencia de ningún framework, que convierte llamadas en bruto de ratón, rueda y gestos en lecturas de retícula, clics de punto, zoom, desplazamiento y, en la edición financiera, comportamiento de herramientas de dibujo y divisor de panel. No tiene ninguna dependencia de VCL, FMX ni de ninguna API de ventanas; una vista reenvía lo que recibe de sus propios eventos de entrada (un TsgcVCLChart, un TsgcFMXChart, o un bucle anfitrión personalizado) como coordenadas de píxel del gráfico, el mismo espacio de píxeles de dispositivo que el rectángulo pasado a **TsgcChart.Render**. Todo gráfico posee exactamente una instancia de interacción, creada en su constructor y expuesta como **TsgcChart.Interaction**; las aplicaciones la configuran ahí en lugar de crear una propia.

Propiedades

Propiedad	Tipo	Descripción
Crosshair	Boolean	Pinta la retícula y las lecturas de los ejes mientras el cursor está sobre el trazado. Valor predeterminado True.
CrosshairColor	TsgcUIColor	0 (el valor predeterminado) usa el color del tema.
Tooltip	Boolean	Muestra la información sobre herramientas del punto más cercano mientras el cursor está encima. Valor predeterminado True.
ZoomMode	TsgcChartZoomMode	`czmNone`, `czmX`, `czmY` o `czmXY`: sobre qué ejes actúan el zoom y el desplazamiento. Valor predeterminado `czmX`.
WheelZoom	Boolean	La rueda hace zoom alrededor del cursor; Mayús + rueda desplaza horizontalmente en su lugar. Valor predeterminado True.
BoxZoom	Boolean	Un arrastre con el botón izquierdo dibuja una banda elástica y hace zoom sobre ella. Valor predeterminado True. Ctrl + arrastre izquierdo siempre desplaza en su lugar, y un arrastre con el botón derecho o central desplaza independientemente de esta propiedad.
Pan	Boolean	Habilita el desplazamiento mediante arrastre. Valor predeterminado True.
AutoScrollResumeDelay	Integer	Milisegundos sin interacción antes de que se restablezca un zoom horizontal y un **TsgcChart.TimeWindow** vuelva a desplazarse; 0 significa que nunca se reanuda automáticamente. Valor predeterminado 3000. Véase Desplazamiento automático.
SnapToPoint	Boolean	La retícula salta al punto más cercano: más cercano en X, y después más cercano en Y entre las series en ese X (si no, el más cercano en distancia de píxeles cuando es False). Valor predeterminado True.
DrawingTool	TsgcChartDrawingTool	Solo edición financiera. `cdtNone` (el valor predeterminado) permite que una pulsación izquierda seleccione y arrastre un dibujo existente; cualquier otro valor (`cdtTrendLine`, `cdtHorizontalLine`, `cdtVerticalLine`, `cdtRectangle`, `cdtFibonacci`) hace que el siguiente arrastre izquierdo en el trazado cree un dibujo de ese tipo (pulsación = primer anclaje, liberación = segundo anclaje), tras lo cual la propiedad vuelve a `cdtNone`.

Paused	Boolean (solo lectura)	True mientras el desplazamiento automático está en pausa por un zoom o un desplazamiento. Véase Desplazamiento automático.
Dragging	Boolean (solo lectura)	True mientras hay en curso un arrastre de zoom de caja, desplazamiento, divisor o dibujo.
DragMode	TsgcChartDragMode (solo lectura)	`cdmNone` , `cdmBox` , `cdmPan` , o, en la edición financiera, `cdmSplitter` (se arrastra un divisor de panel, véase Panes), `cdmDrawCreate` o `cdmDrawMove` (se arrastra un dibujo o uno de sus anclajes).
Chart	TComponent (solo lectura)	El gráfico al que está conectada esta interacción.

Métodos

Método	Devuelve	Descripción
MouseMove(X, Y: Double)		Alimenta un movimiento del puntero, en píxeles del gráfico. Actualiza la retícula, continúa un arrastre en curso, o genera **TsgcChart.OnCrosshairMove** cuando está inactivo sobre el trazado.
MouseDown(X, Y: Double; aButton: TsgcChartMouseButton; aShift: TsgcChartShiftState)		Alimenta la pulsación de un botón. **aButton** es `cmbLeft` , `cmbRight` o `cmbMiddle` ; **aShift** es un conjunto de `cskShift` , `cskCtrl` , `cskAlt` . Inicia un zoom de caja, un desplazamiento, un arrastre de divisor o un arrastre de dibujo según las propiedades anteriores.
MouseUp(X, Y: Double; aButton: TsgcChartMouseButton; aShift: TsgcChartShiftState)		Alimenta la liberación de un botón. Completa el arrastre iniciado por **MouseDown**, o, cuando el puntero nunca superó el umbral de arrastre, resuelve un clic y genera **TsgcChart.OnSeriesClick**.
MouseWheel(X, Y: Double; aDelta: Integer; aShift: TsgcChartShiftState)		Alimenta un paso de rueda. **aDelta** es 120 por paso, positivo alejándose del usuario (acercar el zoom).
MouseLeave		Alimenta la salida del puntero del control, ocultando la retícula y la información sobre herramientas.
DoubleClick(X, Y: Double)		Alimenta un doble clic, en píxeles del gráfico. Llama a **ResetZoom**.
Gesture(aKind: TsgcChartGestureKind; aCenterX, aCenterY, aFactorOrDX, aDY: Double)		Alimenta un gesto táctil. Para `cgZoom` , **aFactorOrDX** es la escala del pellizco (mayor que 1 acerca el zoom) alrededor de **aCenterX, aCenterY**; para `cgPan` , **aFactorOrDX** y **aDY** son el movimiento del dedo en píxeles.
ResetZoom		Restablece el zoom de cada eje y reanuda el desplazamiento automático.
DeleteSelectedDrawing	Boolean	Solo edición financiera. Elimina el dibujo seleccionado de **TsgcChart.Drawings** a menos que esté Locked; una vista la llama para la tecla Supr. True cuando se eliminó un dibujo.

HitTest(X, Y: Double)	TsgcChartHitInfo	Qué hay bajo una posición de píxel del gráfico tras el último **TsgcChart.Render**, sin efecto secundario alguno. Véase HitTest y TsgcChartHitInfo.
CheckAutoScroll	Boolean	Llamado por **TsgcChart.Render** antes de calcular los rangos. Restablece el zoom horizontal una vez transcurrido **AutoScrollResumeDelay** desde la última interacción; True cuando lo hizo. Véase Desplazamiento automático.
DrawOverlay(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Llamado por **TsgcChart.Render** después del contenido del gráfico. Pinta la retícula, la información sobre herramientas y la banda elástica dentro de **aRect** de **aCanvas**.

Eventos

Estos tres eventos se publican en el propio **TsgcChart**, no en TsgcChartInteraction: la interacción es quien los genera en el gráfico al que está conectada, así que una aplicación los asigna en el gráfico. Véase TsgcChart.

Evento	Tipo	Descripción
TsgcChart.OnCrosshairMove	TsgcChartHitEvent	El puntero se movió sobre el trazado estando inactivo (sin ningún arrastre en curso); **aHit** es el resultado de **HitTest** en el puntero.
TsgcChart.OnSeriesClick	TsgcChartHitEvent	Un clic izquierdo cayó dentro del radio de clic de un punto; **aHit.Series** y **aHit.Index** lo identifican.
TsgcChart.OnZoom	TNotifyEvent	Se genera después de un zoom, un desplazamiento interactivo o **ResetZoom**.

TsgcChartHitEvent se declara como:

```
TsgcChartHitEvent = procedure(Sender: TObject; const aHit: TsgcChartHitInfo)
  of object;
```

HitTest y TsgcChartHitInfo

HitTest informa de qué hay en una posición de píxel del gráfico, usando el rectángulo de trazado calculado por el último **TsgcChart.Render**; no cambia nada y es seguro llamarlo en cualquier momento, incluso fuera de **MouseMove**, **MouseDown** o **MouseUp**. Devuelve un registro **TsgcChartHitInfo** con estos campos:

Campo	Tipo	Descripción
Area	TsgcChartHitArea	La parte del gráfico bajo la posición: `chaNone`, `chaPlot`, `chaAxis`, `chaLegend` o `chaTitle`.
Axis	TsgcChartAxis	El eje bajo la posición cuando **Area** es `chaAxis`.
Series	TComponent	La **TsgcChartSeries** del punto más cercano, nil cuando no hay ninguna. Declarada como TComponent porque TsgcChartInteraction se compila antes que TsgcChart y los tipos de serie.
SeriesIndex	Integer	Índice de **Series** en el gráfico, -1 cuando no hay ninguna.
Index	Integer	Índice lógico del punto en el búfer, 0 es el más antiguo, -1 cuando no hay ninguno.

XValue, YValue	Double	Valores del punto. **YValue** es el Close de una vela.
IsOHLC	Boolean	True cuando el punto proviene de una serie OHLC o de velas.
Open, High, Low, Close	Double	Se establecen solo cuando **IsOHLC** es True.
PixelX, PixelY	Double	Posición del punto en píxeles del gráfico.
Distance	Double	Distancia en píxeles entre la posición consultada y el punto; 0 dentro de una barra o una vela.
CursorX, CursorY	Double	La posición consultada, tal como se pasó a **HitTest**.
SubIndex	Integer	Elemento de la serie bajo la posición: un sector, un punto de radar, una celda de mapa de calor, una profundidad de treemap o una etapa de embudo; -1 si no hay ninguno.
ZValue	Double	Tercer valor del elemento (el valor de la celda del mapa de calor), NaN si no hay ninguno.
Percent	Double	Proporción del elemento en su serie (gráfico circular, embudo, treemap), NaN si no hay ninguno.
Text	string	Texto de categoría del elemento (sector, radio, celdas, ruta de nodo o etapa), vacío si no hay ninguno.

TsgcChart.OnSeriesClick solo se dispara cuando **Distance** está dentro de un pequeño radio de clic del punto; un clic que cae en una zona vacía del trazado sigue llamando internamente a **HitTest** pero no genera el evento.

Áreas de trazado: desplazamiento del ratón, clic y acciones de las series

El desplazamiento del ratón, el clic y las informaciones sobre herramientas llegan a las series de todos los modos de área de trazado: celdas de gráfico circular, embudo y treemap, áreas polares de radar, y el mapa de calor en un área XY. El zoom, el desplazamiento, la retícula y el restablecimiento del zoom con doble clic solo funcionan en áreas XY. No hay eventos de gráfico nuevos: el desplazamiento del ratón dispara **OnCrosshairMove** y un clic dispara **OnSeriesClick**, ambos con los campos de impacto anteriores; después de **OnSeriesClick** la serie ejecuta su propia acción de clic mediante **AreaClick**, de modo que un gráfico circular con **ExplodeOnClick** separa el sector y un treemap con **DrillOnClick** profundiza en el grupo. Una serie puede proporcionar su propio texto de información sobre herramientas, que se dibuja como líneas simples. Con un mapa de calor bajo una serie de líneas, un punto de línea dentro del radio de clic prevalece, y en caso contrario prevalece la celda del mapa de calor.

```
uses
  SysUtils,
  sgcUICore_Theme, sgcUICore_Canvas, sgcUICore_Canvas_Raster,
  sgcCharts_Chart, sgcCharts_Interaction, sgcCharts_Series_Category,
  sgcCharts_Series_Pie;

var
  vChart: TsgcChart;
  vPie: TsgcChartSeries_Pie;
  vRaster: TsgcUICanvas_Raster;
  vRect: TsgcUIRect;
  vHit: TsgcChartHitInfo;
  vX, vY: Double;
begin
  vChart := TsgcChart.Create(nil);
```

```
  vRaster := TsgcUICanvas_Raster.Create(400, 400);
  vPie := TsgcChartSeries_Pie.Create(nil);
  try
    vPie.Chart := vChart;
    vPie.ExplodeOnClick := True;
    vPie.SetValue('Web', 50);
    vPie.SetValue('Store', 30);
    vPie.SetValue('Phone', 20);
    // the hit test uses the geometry of the last render
    vChart.Render(vRaster, sgcUIRect(0, 0, 400, 400));

    // a point right of 12 o'clock: inside the first slice
    vRect := vChart.SeriesAreaRect(vPie);
    vX := (vRect.Left + vRect.Right) / 2 + 20;
    vY := (vRect.Top + vRect.Bottom) / 2 - 40;
    vHit := vChart.Interaction.HitTest(vX, vY);
    if vHit.Series = vPie then
      Writeln('slice ', vHit.SubIndex, ' ', vHit.Text, ': ',
        FormatFloat('0.#', vHit.Percent), '%');

    // a click: OnSeriesClick fires, then the pie explodes the slice
    vChart.Interaction.MouseDown(vX, vY, cmbLeft, []);
    vChart.Interaction.MouseUp(vX, vY, cmbLeft, []);
    Writeln('exploded: ',
      TsgcChartPieSlice(vPie.Slices[vHit.SubIndex]).Exploded);
  finally
    vPie.Free;
    vRaster.Free;
    vChart.Free;
  end;
end.
```

Desplazamiento automático

Un gráfico con un **TsgcChart.TimeWindow** distinto de cero normalmente mantiene su eje horizontal mostrando los datos más recientes. Hacer zoom o desplazar los ejes horizontales congela ese comportamiento para que el rango con zoom permanezca en pantalla mientras siguen llegando datos nuevos; **Paused** informa True mientras esto está en efecto. No hay ningún método independiente de pausa o reanudación: **TsgcChart.Render** llama a **CheckAutoScroll** en cada render, y una vez que han transcurrido **AutoScrollResumeDelay** milisegundos sin interacción, el zoom horizontal se restablece y la ventana de tiempo vuelve a desplazarse; **AutoScrollResumeDelay** establecido en 0 desactiva la reanudación automática, dejando **ResetZoom** como única forma de volver. No interviene ningún temporizador, la comprobación solo ocurre en el momento del render.

Ejemplo: alimentar entrada sintética y leer el impacto

TsgcChartInteraction recibe coordenadas simples, así que no se necesita ningún formulario VCL ni FMX para ejercitarlo, un programa de consola puede manejarlo directamente una vez que el gráfico se ha renderizado al menos una vez (para que se conozca **PlotRect**).

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster, sgcCharts_Interaction;

type
  THandler = class
    procedure OnCrosshair(Sender: TObject; const aHit: TsgcChartHitInfo);
    procedure OnClick(Sender: TObject; const aHit: TsgcChartHitInfo);
  end;

procedure THandler.OnCrosshair(Sender: TObject; const aHit: TsgcChartHitInfo);
begin
  Writeln('crosshair: seriesIndex=', aHit.SeriesIndex, ' index=', aHit.Index,
    ' x=', aHit.XValue: 0: 2, ' y=', aHit.YValue: 0: 2);
```

```
end;

procedure THandler.OnClick(Sender: TObject; const aHit: TsgcChartHitInfo);
begin
  if aHit.Series <> nil then
    Writeln('click: series=', TsgcChartSeries(aHit.Series).Name, ' index=',
      aHit.Index)
  else
    Writeln('click: no point near cursor');
end;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  vRaster: TsgcUICanvas_Raster;
  vHandler: THandler;
  vHit: TsgcChartHitInfo;
  vAxisX, vAxisY: TsgcChartAxis;
  vClickX, vClickY: Double;
  i: Integer;
begin
  vChart := TsgcChart.Create(nil);
  vHandler := THandler.Create;
  try
    // TsgcChart already owns a bottom and a left axis, and Interaction
    // (created automatically, as vChart.Interaction)
    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Name := 'Signal';
      vSeries.Chart := vChart;
      for i := 0 to 99 do
        vSeries.AddXY(i, Sin(i / 10) * 50 + 50);

      vChart.OnCrosshairMove := vHandler.OnCrosshair;
      vChart.OnSeriesClick := vHandler.OnClick;

      vRaster := TsgcUICanvas_Raster.Create(400, 300);
      try
        vChart.Scale := 1;
        vChart.Render(vRaster, sgcUIRect(0, 0, 400, 300));

        // hover: fires OnCrosshairMove through the handler above
        vChart.Interaction.MouseMove(200, 150);

        // HitTest reports the point nearest to a chart pixel position
        vHit := vChart.Interaction.HitTest(200, 150);
        Writeln('HitTest: seriesIndex=', vHit.SeriesIndex, ' index=',
          vHit.Index, ' x=', vHit.XValue: 0: 2, ' y=', vHit.YValue: 0: 2);

        // click exactly on that point: fires OnSeriesClick
        vAxisX := vSeries.XAxis;
        vAxisY := vSeries.YAxis;
        vClickX := vAxisX.ValueToPixel(vHit.XValue, vChart.PlotRect.Left,
          vChart.PlotRect.Right - vChart.PlotRect.Left);
        vClickY := vAxisY.ValueToPixel(vHit.YValue, vChart.PlotRect.Top,
          vChart.PlotRect.Bottom - vChart.PlotRect.Top);
        vChart.Interaction.MouseDown(vClickX, vClickY, cmbLeft, []);
        vChart.Interaction.MouseUp(vClickX, vClickY, cmbLeft, []);
      finally
        vRaster.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vHandler.Free;
    vChart.Free;
```

```
  end;
end.
```

Esto imprime el impacto de la retícula, el resultado equivalente de **HitTest**, y la serie y el índice resueltos por el clic, por ejemplo:

```
crosshair: seriesIndex=0 index=46 x=46.00 y=0.32
HitTest: seriesIndex=0 index=46 x=46.00 y=0.32
click: series=Signal index=46
```

Véase también

- TsgcChart
- Axes
- TsgcChartSeries_Line
- TsgcVCLChart
- TsgcFMXChart
- Panes
- Áreas de trazado

TsgcChartDataSource

Unidad: sgcCharts_DataSource
Clase base de todo origen de datos de gráfico. Añade puntos a un **Buffer** (TsgcChartBuffer) y velas a un **OHLCBuffer** (**TsgcChartOHLCBuffer**). Ninguno de los dos es propiedad del origen de datos, se establecen en código. Cada adición pasa por el propio bloqueo del búfer, de modo que un hilo de alimentación puede añadir mientras el hilo de renderizado lee una instantánea.
Mapping indica al origen de datos dónde están los valores X, Y y OHLC en cada registro entrante: una ruta JSON para la entrada JSON, un nombre de columna o un índice de columna basado en 0 para la entrada CSV, o un nombre de campo para un conjunto de datos. Un **XPath** vacío hace que X sea el índice de registro basado en 0. Un registro produce un punto cuando **Buffer** está asignado y **YPath** está establecido, y una vela cuando **OHLCBuffer** está asignado y **OpenPath**, **HighPath**, **LowPath** y **ClosePath** están establecidos (**VolumePath** es opcional; un volumen ausente es 0). Los registros cuyo X o valores no se pueden leer o convertir se omiten.
Mapping.TimeFormat convierte X: `ctfNumber` conserva el número, `ctfUnixSeconds` y `ctfUnixMilliseconds` convierten un tiempo Unix en un `TDateTime` UTC, `ctfISO8601` interpreta un texto ISO 8601 (un desplazamiento `Z` o `+hh:mm` se convierte a UTC; un texto sin zona se conserva tal cual) y conserva un número simple como `TDateTime`. Los números siempre se interpretan con . como separador decimal, sea cual sea la configuración regional.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Active	Boolean	True	False hace que el origen de datos ignore cada adición.
Mapping	TsgcChartDataMapping	—	XPath, YPath, OpenPath, HighPath, LowPath, ClosePath, VolumePath, LabelPath y TimeFormat. Véase más arriba, y Series de categorías y mapas de calor más abajo para **LabelPath**.
Throttle	Integer	0	Milisegundos entre notificaciones de **OnDataChanged**. Véase más abajo.
OnBeforeAppend	TsgcChartBeforeAppendEvent	nil	Se dispara antes de añadir un punto, con `aX` y `aY` modificables y `aAccept` establecido en True; establecer `aAccept` en False omite el punto.
OnBeforeAppendOHLC	TsgcChartBeforeAppendOHLCEvent	nil	Se dispara antes de añadir una vela, con `aX`, `aOpen`, `aHigh`, `aLow`, `aClose`, `aVolume` modificables y `aAccept` establecido en True; establecer `aAccept` en False omite la vela.
OnDataChanged	TNotifyEvent	nil	Se dispara después de añadir uno o más puntos o velas. Véase Notificación más abajo.

Métodos

Método	Descripción
BeginUpdate / EndUpdate	Difiere las notificaciones hasta el **EndUpdate** correspondiente, que entonces envía una única notificación (regulada) cuando se ha añadido algo mientras tanto. Se puede anidar libremente; solo el **EndUpdate** más externo puede dispararse.
Clear	Vacía **Buffer** y **OHLCBuffer**, vacía una serie de categorías o un mapa de calor enlazados, y restablece el índice de registro utilizado cuando **XPath** está vacío.

Series de categorías y mapas de calor

La propiedad **Series** enlaza una serie X/Y o de velas, y también una serie de categorías (gráfico circular, radar, embudo, treemap) o un mapa de calor. **Mapping.LabelPath** es la ruta JSON o la columna CSV del texto del elemento. Un registro de categoría con etiqueta llama a **SetValue**(etiqueta, Y), de modo que el último valor de un texto prevalece; sin una asignación de etiqueta establece el elemento en X redondeado al índice más cercano (X es el índice del registro cuando **XPath** está vacío). Un registro cuya etiqueta está asignada pero falta o está vacía no escribe ningún elemento, y **OnBeforeAppend** sigue filtrando (X, Y). Un registro de mapa de calor añade una línea: un registro JSON cuyo **YPath** es una matriz (la clave es X cuando **XPath** está establecido, y si no el contador de líneas en curso), o las celdas de un registro CSV; los valores que no son números son celdas nulas. Una carga escribe la serie enlazada dentro de su **BeginUpdate** / **EndUpdate**, de modo que se repinta una sola vez, y un origen de datos nunca escribe en una serie en tiempo de diseño, de modo que los datos en vivo nunca llegan a un DFM. Los orígenes de conjunto de datos y de tabla usan **LabelField** y **LabelColumn** en lugar de **LabelPath**.

Notificación e hilos

OnDataChanged está regulado: tras una adición, el evento se dispara de inmediato cuando la última notificación tiene al menos **Throttle** ms de antigüedad; en caso contrario, un hilo auxiliar lo dispara cuando se cierra la ventana de regulación. Nunca hay más de una notificación pendiente, de modo que una ráfaga de adiciones produce un solo evento por ventana. **Throttle** = 0 notifica en cada adición (aun así fusionadas mientras hay una notificación pendiente).
Delphi 2010 y posteriores entregan **OnDataChanged** en el hilo principal, mediante `TThread.Queue`; una aplicación de consola o un servicio deben llamar a `CheckSynchronize` para recibirlo. Delphi 7 a 2009 no tienen `TThread.Queue`, por lo que allí el evento se dispara en el hilo que hizo la adición, o en el hilo de regulación, y el controlador debe sincronizarse a sí mismo antes de tocar un control.

Descendientes

- TsgcChartDataSource_Manual: se añade directamente desde código.
- TsgcChartDataSource_DataSet: carga desde un `TDataSet`.
- TsgcChartDataSource_JSON: carga desde un documento JSON.
- TsgcChartDataSource_CSV: carga desde texto CSV.
- TsgcChartDataSource_Table: carga desde una tabla que implementa `IsgcUITable`, como un modelo de datos de sgcGrid.
- Los orígenes de datos de puente, alimentados por una conexión sgcWebSockets en vivo.

TsgcChartDataSource_Manual

Unidad: sgcCharts_DataSource
El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Mapping**, **Active**, **Throttle**, **BeginUpdate**/**EndUpdate**, **Clear** y los eventos previos a la adición y de cambio de datos. No añade ninguna propiedad nueva, solo los tres métodos que una aplicación llama directamente cuando ya tiene los valores en la mano y no necesita que **Mapping** los busque en un registro.

Métodos

Método	Devuelve	Descripción
AppendValue(const aX, aY: Double)	Boolean	Añade un punto a **Buffer**. False cuando el origen de datos está inactivo, **Buffer** es nil, o **OnBeforeAppend** rechazó el punto.
AppendOHLC(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)	Boolean	Añade una vela a **OHLCBuffer**. False cuando el origen de datos está inactivo, **OHLCBuffer** es nil, o **OnBeforeAppendOHLC** rechazó la vela.
AppendJSON(const aJSON: string)	Boolean	Lee un objeto JSON, o una matriz de objetos o matrices, a través de **Mapping**, de la misma manera que TsgcChartDataSource_JSON lee un documento. Devuelve False ante un JSON con formato incorrecto; nunca genera un error por ello.
AppendCategory(const aText: string; aValue: Double)	Boolean	Escribe el valor de **aText** en la serie de categorías enlazada (un upsert; la X de **OnBeforeAppend** es el índice del elemento). False cuando el origen de datos está inactivo, no hay ninguna serie de categorías enlazada, o **OnBeforeAppend** la rechazó.
AppendLine(aKey: Double; const aValues: array of Double)	Boolean	Añade una línea en **aKey** al mapa de calor enlazado. False cuando el origen de datos está inactivo o no hay ningún mapa de calor enlazado.

Ejemplo: Adición de puntos desde código

```
var
  vBuffer: TsgcChartBuffer;
  vSource: TsgcChartDataSource_Manual;
  i: Integer;
begin
  vBuffer := TsgcChartBuffer.Create(1024, cbmOverwrite);
  try
    vSource := TsgcChartDataSource_Manual.Create(nil);
    try
      vSource.Buffer := vBuffer;
      for i := 0 to 99 do
        vSource.AppendValue(i, Sin(i / 10) * 100);
    finally
      vSource.Free;
    end;
  finally
    vBuffer.Free;
```

```
  end;
end;
```

Ejemplo: alimentar un gráfico circular y un mapa de calor

```
uses
  SysUtils,
  sgcCharts_Chart, sgcCharts_DataSource, sgcCharts_Series_Category,
  sgcCharts_Series_Pie, sgcCharts_Series_Heatmap;

var
  vPie: TsgcChartSeries_Pie;
  vHeat: TsgcChartSeries_Heatmap;
  vPieSource, vHeatSource: TsgcChartDataSource_Manual;
  i: Integer;
begin
  vPie := TsgcChartSeries_Pie.Create(nil);
  vHeat := TsgcChartSeries_Heatmap.Create(nil);
  vPieSource := TsgcChartDataSource_Manual.Create(nil);
  vHeatSource := TsgcChartDataSource_Manual.Create(nil);
  try
    // a category series: AppendCategory writes the value of a text
    vPieSource.Series := vPie;
    vPieSource.AppendCategory('Chrome', 64);
    vPieSource.AppendCategory('Safari', 19);
    vPieSource.AppendCategory('Chrome', 66); // replaces 64
    // JSON records: Mapping.LabelPath names the item text
    vPieSource.Mapping.LabelPath := 'browser';
    vPieSource.Mapping.YPath := 'share';
    vPieSource.AppendJSON('[{"browser":"Edge","share":5},' +
      '{"browser":"Firefox","share":3}]');
    for i := 0 to vPie.ItemCount - 1 do
      Writeln(vPie.Slices[i].Text, ' = ', FormatFloat('0', vPie.Slices[i].Value));

    // a heatmap: AppendLine adds one line with its key
    vHeatSource.Series := vHeat;
    vHeatSource.AppendLine(0, [1, 2, 3]);
    vHeatSource.AppendLine(1, [4, 5, 6]);
    Writeln('line 1, cell 2 = ', FormatFloat('0', vHeat.Cells[1, 2]));
  finally
    vHeatSource.Free;
    vPieSource.Free;
    vHeat.Free;
    vPie.Free;
  end;
end.
```

Véase También

- TsgcChartDataSource
- TsgcChartBuffer

TsgcChartDataSource_DataSet

Unidad: sgcCharts_DataSource_DataSet
Carga datos de gráfico desde las filas de un `TDataSet`. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Active**, **Throttle** y los eventos previos a la adición y de cambio de datos; no utiliza la propiedad **Mapping** heredada, ya que los campos se nombran directamente.
XField, **YField** y de **OpenField** a **VolumeField** nombran los campos. Un **XField** vacío hace que X sea el índice de registro basado en 0. Un campo de fecha y hora aporta su `TDateTime` como X; un campo numérico se convierte mediante **Mapping.TimeFormat** (segundos o milisegundos Unix); un campo de texto se interpreta como texto CSV. Una fila con un X o Y nulo no produce ningún punto, una fila con un open, high, low o close nulo no produce ninguna vela, y un volumen nulo es 0.

Propiedades

Propiedad	Tipo	Descripción
DataSet	TDataSet	El conjunto de datos del que se cargan las filas.
XField	string	Campo que contiene X. Vacío significa el índice de registro basado en 0.
YField	string	Campo que contiene Y.
OpenField, HighField, LowField, CloseField	string	Campos que contienen los valores OHLC. Los cuatro deben estar establecidos para que una fila produzca una vela.
VolumeField	string	Campo que contiene el volumen. Opcional; un campo nulo o no establecido es 0.
LabelField	string	Campo del texto del elemento de una serie de categorías enlazada (gráfico circular, radar, embudo, treemap): cada fila llama a **SetValue**(texto, Y), y una fila con un texto nulo o vacío no escribe ningún elemento. Sin **LabelField** cada fila establece el elemento en su X (el índice del registro cuando **XField** está vacío). Véase Series de categorías y mapas de calor.

Métodos

Método	Descripción
LoadFromDataSet	Añade cada fila del **DataSet** abierto, con los controles deshabilitados y el registro actual restaurado después, y envía una única notificación al final. No hace nada cuando **DataSet** es nil o está cerrado.
Refresh	Vacía los búferes y vuelve a llamar a **LoadFromDataSet**. No hace nada cuando **DataSet** es nil o está cerrado.

Ejemplo: Carga de un búfer desde un conjunto de datos

```
var
  vBuffer: TsgcChartBuffer;
  vSource: TsgcChartDataSource_DataSet;
  vDataSet: TClientDataSet;
begin
```

```
  vBuffer := TsgcChartBuffer.Create(4096, cbmOverwrite);
  try
    vDataSet := TClientDataSet.Create(nil);
    try
      vDataSet.FieldDefs.Add('ts', ftFloat);
      vDataSet.FieldDefs.Add('price', ftFloat);
      vDataSet.CreateDataSet;
      vDataSet.Open;
      vDataSet.AppendRecord([1.0, 100.5]);
      vDataSet.AppendRecord([2.0, 101.25]);

      vSource := TsgcChartDataSource_DataSet.Create(nil);
      try
        vSource.Buffer := vBuffer;
        vSource.DataSet := vDataSet;
        vSource.XField := 'ts';
        vSource.YField := 'price';
        vSource.LoadFromDataSet;
      finally
        vSource.Free;
      end;
    finally
      vDataSet.Free;
    end;
  finally
    vBuffer.Free;
  end;
end;
```

Véase También

- TsgcChartDataSource
- TsgcChartBuffer

TsgcChartDataSource_JSON

Unidad: sgcCharts_DataSource_JSON
Carga datos de gráfico desde un documento JSON. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Mapping**, **Active**, **Throttle** y los eventos previos a la adición y de cambio de datos.
ArrayPath selecciona los registros; un **ArrayPath** vacío es la raíz del documento. Una matriz de objetos o matrices produce un registro por elemento, un objeto produce un registro. Cada registro se lee entonces a través de **Mapping**, exactamente como se describe en TsgcChartDataSource. Las cargas se añaden a los búferes, así que llame primero a **Clear** para sustituir los datos en lugar de añadirlos, y envían una única notificación al final. Un JSON con formato incorrecto o un **ArrayPath** ausente generan `EsgcChartJSONError`.

Propiedades

Propiedad	Tipo	Descripción
ArrayPath	string	Ruta a la matriz (u objeto) de registros dentro del documento. Vacío significa la raíz del documento.

Métodos

Método	Descripción
LoadFromString(const aJSON: string)	Carga desde una cadena JSON.
LoadFromStream(aStream: TStream)	Carga desde un flujo de texto UTF-8; una BOM es opcional.
LoadFromFile(const aFileName: string)	Carga desde un archivo UTF-8; una BOM es opcional.

Ejemplo: Carga de velas desde una matriz JSON

```
var
  vOHLC: TsgcChartOHLCBuffer;
  vSource: TsgcChartDataSource_JSON;
const
  CS_JSON = '[{"t":1700000000,"o":100,"h":105,"l":99,"c":103,"v":10}]';
begin
  vOHLC := TsgcChartOHLCBuffer.Create(4096, cbmOverwrite);
  try
    vSource := TsgcChartDataSource_JSON.Create(nil);
    try
      vSource.OHLCBuffer := vOHLC;
      vSource.Mapping.XPath := 't';
      vSource.Mapping.TimeFormat := ctfUnixSeconds;
      vSource.Mapping.OpenPath := 'o';
      vSource.Mapping.HighPath := 'h';
      vSource.Mapping.LowPath := 'l';
      vSource.Mapping.ClosePath := 'c';
      vSource.Mapping.VolumePath := 'v';
      vSource.LoadFromString(CS_JSON);
    finally
      vSource.Free;
    end;
  finally
    vOHLC.Free;
```

```
  end;
end;
```

Véase También

- TsgcChartDataSource
- TsgcChartDataSource_Manual (AppendJSON)
- TsgcChartBuffer

TsgcChartDataSource_CSV

Unidad: sgcCharts_DataSource_CSV
Carga datos de gráfico desde texto CSV (RFC 4180). El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Mapping**, **Active**, **Throttle** y los eventos previos a la adición y de cambio de datos.
Los campos se separan con **Delimiter**; un campo puede ir entre comillas con ", y un campo entre comillas puede contener el delimitador, saltos de línea y comillas dobles. Las líneas terminan con CRLF, LF o CR; las líneas vacías se omiten. Cuando **HasHeader** es True, el primer registro nombra las columnas.
Las rutas de **Mapping** seleccionan las columnas: un nombre de encabezado (sin distinguir mayúsculas y minúsculas) o un índice de columna basado en 0. Los números utilizan **DecimalSeparator** (. de forma predeterminada) sea cual sea la configuración regional. Una fila cuyos valores asignados faltan o no son numéricos se omite. Las cargas se añaden a los búferes (llame primero a **Clear** para sustituir los datos) y envían una única notificación al final.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Delimiter	Char	,	Separador de campos.
HasHeader	Boolean	True	Cuando es True, el primer registro nombra las columnas y las rutas de Mapping pueden usar esos nombres en lugar de un índice de columna.
DecimalSeparator	Char	.	Separador decimal utilizado para interpretar los campos numéricos, sea cual sea la configuración regional.

Métodos

Método	Descripción
LoadFromString(const aText: string)	Carga desde una cadena CSV.
LoadFromStream(aStream: TStream)	Carga desde un flujo de texto UTF-8; una BOM es opcional.
LoadFromFile(const aFileName: string)	Carga desde un archivo UTF-8; una BOM es opcional.

Ejemplo: Carga de puntos desde un archivo CSV

```
var
  vBuffer: TsgcChartBuffer;
  vSource: TsgcChartDataSource_CSV;
begin
  vBuffer := TsgcChartBuffer.Create(8192, cbmOverwrite);
  try
    vSource := TsgcChartDataSource_CSV.Create(nil);
    try
      vSource.Buffer := vBuffer;
      vSource.HasHeader := True;
      vSource.Mapping.XPath := 'time';
      vSource.Mapping.YPath := 'value';
      vSource.LoadFromFile('sensor.csv');
    finally
      vSource.Free;
```

```
    end;
  finally
    vBuffer.Free;
  end;
end;
```

Véase También

- TsgcChartDataSource
- TsgcChartBuffer

TsgcChartDataSource_Table

Unidad: sgcCharts_DataSource_Table
Carga datos del gráfico desde cualquier tabla que implemente la interfaz `IsgcUITable` de sgcUICore (unidad `sgcUICore_Table`), por ejemplo el modelo de datos de una cuadrícula sgcGrid, sin dependencia entre ambos productos. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **Series**, **Active**, **Throttle** y los eventos antes de añadir y de cambio de datos; no usa las rutas heredadas de **Mapping**, ya que las columnas se nombran directamente.
XColumn y **YColumn** nombran las columnas, comparadas sin distinguir mayúsculas de minúsculas con los nombres de columna de la tabla (en sgcGrid, el **FieldName** de la columna). Un **XColumn** vacío hace que X sea el índice de fila basado en 0. Un valor de fecha/hora da su `TDateTime` como X, un valor numérico se convierte con **Mapping.TimeFormat** y un valor de texto se analiza como texto CSV. Una fila con X o Y vacío o no convertible no genera ningún punto. Las filas se leen en el orden de la tabla, que en sgcGrid es la vista ordenada y filtrada.
El origen registra un listener en la tabla. Cada cambio de la tabla recarga la tabla completa cuando **Active** es True y la versión de la tabla ha cambiado desde la última carga. La recarga se ejecuta en el hilo que modificó la tabla, y la notificación al gráfico se entrega después como en los demás orígenes de datos. Una tabla enlazada con **SetTable** debe desenlazarse con `SetTable(nil)` antes de liberarla; un componente **Table** se desenlaza automáticamente.

Propiedades

Propiedad	Tipo	Descripción
Table	TComponent	Un componente que implementa `IsgcUITable`. Asignar un componente que no lo implementa provoca una excepción.
XColumn	string	Columna que contiene X. Vacío significa el índice de fila basado en 0.
YColumn	string	Columna que contiene Y.
LabelColumn	string	Columna del texto del elemento de una serie de categorías enlazada, o varias columnas separadas por `;`, cuyos textos se unen con el **PathDelimiter** de un treemap (de modo que una tabla con columnas Region, Country y City alimenta un treemap de tres niveles) y con `/` en los demás casos; las partes finales vacías se descartan. Para un mapa de calor, las etiquetas de las líneas.

Métodos

Método	Descripción
SetTable	Enlaza cualquier objeto que implemente `IsgcUITable`, por ejemplo el modelo **Data** de una cuadrícula sgcGrid. nil lo desenlaza.
GetTable	Devuelve la tabla enlazada, nil si no hay ninguna.
LoadFromTable	Añade todas las filas de la tabla, manteniendo el bloqueo de la tabla mientras lee, y envía una notificación al final.
Refresh	Vacía los búferes y vuelve a cargar la tabla. No hace nada si no hay ninguna tabla enlazada.

Series de categorías y mapas de calor

Con una serie de categorías cada fila llama a **SetValue**(etiqueta, Y), y una fila con una etiqueta vacía no escribe ningún elemento; sin **LabelColumn** cada fila establece el elemento en su X (el índice de la fila cuando **XColumn** está vacío). Una recarga se ejecuta dentro de **BeginReload** / **EndReload** de la serie, de modo que los elementos conservan sus colores e indicadores y los elementos de las filas eliminadas desaparecen. Con un mapa de calor cada carga sustituye la matriz por una línea por fila (la clave es X cuando **XColumn** está establecido, y si no el índice de la fila). Las celdas son las columnas de **YColumn** separadas por ; o, cuando **YColumn** está vacío, todas las columnas numéricas excepto **XColumn** y las columnas de etiqueta; las etiquetas de las celdas son los nombres de las columnas. Esta es la vía para el resultado de una tabla dinámica o una cuadrícula.

Ejemplo: un gráfico circular alimentado desde una tabla

```
uses
  Classes, SysUtils, Variants,
  sgcUICore_Table, sgcCharts_Chart, sgcCharts_DataSource_Table,
  sgcCharts_Series_Category, sgcCharts_Series_Pie;

type
  { a small table with two columns, Region (text) and Sales (number); a
    live table also keeps its listeners and calls them after a change }
  TSalesTable = class(TComponent, IsgcUITable)
  private
    FRegions: array of string;
    FSales: array of Double;
  public
    procedure AddRow(const aRegion: string; aSales: Double);
    // IsgcUITable
    function TableRowCount: Integer;
    function TableColCount: Integer;
    function TableColName(aCol: Integer): string;
    function TableValue(aRow, aCol: Integer): Variant;
    function TableText(aRow, aCol: Integer): string;
    procedure TableLock;
    procedure TableUnlock;
    function TableVersion: Cardinal;
    procedure TableAddListener(aListener: TNotifyEvent);
    procedure TableRemoveListener(aListener: TNotifyEvent);
  end;

procedure TSalesTable.AddRow(const aRegion: string; aSales: Double);
begin
  SetLength(FRegions, Length(FRegions) + 1);
  SetLength(FSales, Length(FSales) + 1);
  FRegions[High(FRegions)] := aRegion;
  FSales[High(FSales)] := aSales;
end;

function TSalesTable.TableRowCount: Integer;
begin
  Result := Length(FRegions);
end;

function TSalesTable.TableColCount: Integer;
begin
  Result := 2;
end;

function TSalesTable.TableColName(aCol: Integer): string;
begin
  if aCol = 0 then
    Result := 'Region'
  else
    Result := 'Sales';
end;
```

```
function TSalesTable.TableValue(aRow, aCol: Integer): Variant;
begin
  if aCol = 0 then
    Result := FRegions[aRow]
  else
    Result := FSales[aRow];
end;

function TSalesTable.TableText(aRow, aCol: Integer): string;
begin
  if aCol = 0 then
    Result := FRegions[aRow]
  else
    Result := FloatToStr(FSales[aRow]);
end;

procedure TSalesTable.TableLock;
begin
end;

procedure TSalesTable.TableUnlock;
begin
end;

function TSalesTable.TableVersion: Cardinal;
begin
  Result := Length(FRegions);
end;

procedure TSalesTable.TableAddListener(aListener: TNotifyEvent);
begin
end;

procedure TSalesTable.TableRemoveListener(aListener: TNotifyEvent);
begin
end;

var
  vTable: TSalesTable;
  vPie: TsgcChartSeries_Pie;
  vSource: TsgcChartDataSource_Table;
  i: Integer;
begin
  vTable := TSalesTable.Create(nil);
  vPie := TsgcChartSeries_Pie.Create(nil);
  vSource := TsgcChartDataSource_Table.Create(nil);
  try
    vTable.AddRow('North', 420);
    vTable.AddRow('South', 310);
    vTable.AddRow('East', 275);

    vSource.Series := vPie;
    vSource.LabelColumn := 'Region'; // the item texts
    vSource.YColumn := 'Sales'; // the values
    vSource.Table := vTable;
    vSource.Refresh;

    for i := 0 to vPie.ItemCount - 1 do
      Writeln(vPie.Slices[i].Text, ' = ', FormatFloat('0', vPie.Slices[i].Value));
  finally
    vSource.Free;
    vPie.Free;
    vTable.Free;
  end;
end.
```

Véase también

- TsgcChartDataSource
- TsgcChartSeries_Pie
- TsgcChartSeries_Heatmap

Descripción General del Puente

El paquete `sgcChartsWS` añade orígenes de datos de gráfico alimentados por una conexión sgcWebSockets en vivo. Es un paquete independiente del paquete principal `sgcCharts` y requiere sgcWebSockets para compilar; un proyecto que solo utiliza los orígenes de datos principales (TsgcChartDataSource y sus descendientes manual, de conjunto de datos, JSON y CSV) no necesita sgcWebSockets en absoluto.

Componentes

- TsgcChartDataSource_MQTT: representa en el gráfico las publicaciones MQTT recibidas por un `TsgcWSPClient_MQTT`.
- TsgcChartDataSource_WebSocket: representa en el gráfico los mensajes JSON recibidos por un `TsgcWebSocketClient`.
- TsgcChartDataSource_SSE: representa en el gráfico los eventos enviados por el servidor (Server-Sent Events) recibidos por un `TsgcHTTP1Client`.
- TsgcChartDataSource_Binance: se suscribe y representa en el gráfico un flujo de datos de mercado público de Binance a través de un `TsgcWSAPI_Binance`.

Encadenamiento de controladores

Cada origen de datos de puente funciona de la misma manera: mientras está **Active** y su propiedad de cliente está asignada, instala su propio controlador de eventos en el cliente (`OnMQTTPublish`, `OnMessage`, `OnSSEMessage`, o el evento de mensaje del cliente de Binance) y conserva el controlador que hubiera antes. Cada mensaje entrante se representa primero en el gráfico y luego se pasa a ese controlador anterior, de modo que el controlador propio de la aplicación se sigue ejecutando exactamente igual que si el origen de datos de gráfico no estuviera presente.
Establecer **Active** en False, vaciar la propiedad de cliente o destruir el origen de datos restaura el controlador que hubiera antes, a menos que la aplicación haya reemplazado el controlador de nuevo mientras tanto. Este enganche ocurre únicamente en tiempo de ejecución, nunca en tiempo de diseño, de modo que una aplicación que asigna su propio controlador después de adjuntar el origen de datos debe asignarlo antes de establecer la propiedad de cliente (o antes de alternar **Active**) para conservar ambos controladores.
Los mensajes llegan en el hilo que utilice el cliente subyacente para notificar. Las adiciones pasan por el bloqueo y la notificación regulada del TsgcChartDataSource base, por lo que cualquier modo de notificación es seguro de usar con un origen de datos de puente.

TsgcChartDataSource_MQTT

Unidad: sgcChartsWS_DataSource_MQTT
Origen de datos de gráfico alimentado por un cliente MQTT de sgcWebSockets. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Mapping**, **Throttle** y los eventos previos a la adición y de cambio de datos. Consulte Descripción General del Puente para saber cómo funciona el encadenamiento de controladores.
TopicFilter selecciona las publicaciones que se representan en el gráfico, con los comodines MQTT `+` (un nivel) y `#` (este nivel y todos los inferiores); un **TopicFilter** vacío representa todas las publicaciones.
PayloadFormat con `cpfJSON` lee la carga útil a través de **Mapping** (un objeto JSON o una matriz de registros); `cpfNumber` interpreta la carga útil como un número (Y, con . como separador decimal sea cual sea la configuración regional) y utiliza `Now` como X.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Active	Boolean	True	Instala o retira el enganche en **Client**.
Client	TsgcWSPClient_MQTT	nil	El cliente MQTT cuyo `OnMQTTPublish` se engancha.
PayloadFormat	TsgcChartMQTTPayloadFormat	cpfJSON	`cpfJSON` o `cpfNumber`. Véase más arriba.
TopicFilter	string	''	Filtro de tema MQTT (`+`, `#`). Vacío representa todas las publicaciones.

Métodos

Método	Devuelve	Descripción
ProcessPublish(const aTopic, aText: string)	Boolean	Representa en el gráfico una publicación como si procediera de **Client**. False cuando el tema no coincide con **TopicFilter** o la carga útil no se puede leer.

Ejemplo: Representación en el gráfico de lecturas de sensores desde MQTT

La propiedad **Client** recibe el componente de protocolo `TsgcWSPClient_MQTT`, que a su vez está conectado al cliente de transporte (aquí un `TsgcWebSocketClient`) mediante su propia propiedad **Client**, de la misma manera que se conecta cualquier cliente MQTT de sgcWebSockets.

```
var
  vBuffer: TsgcChartBuffer;
  vSource: TsgcChartDataSource_MQTT;
  vWSClient: TsgcWebSocketClient;
  vMQTT: TsgcWSPClient_MQTT;
begin
  vBuffer := TsgcChartBuffer.Create(4096, cbmOverwrite);
  try
    vWSClient := TsgcWebSocketClient.Create(nil);
```

```
    try
      vMQTT := TsgcWSPClient_MQTT.Create(nil);
      try
        vMQTT.Client := vWSClient;
        vWSClient.Host := 'broker.example.com';
        vWSClient.Port := 15675;

        vSource := TsgcChartDataSource_MQTT.Create(nil);
        try
          vSource.Buffer := vBuffer;
          vSource.Client := vMQTT;
          vSource.TopicFilter := 'sensors/+/temperature';
          vSource.PayloadFormat := cpfJSON;
          vSource.Mapping.XPath := 'ts';
          vSource.Mapping.TimeFormat := ctfUnixSeconds;
          vSource.Mapping.YPath := 'value';

          vWSClient.Active := True;
          vMQTT.Subscribe('sensors/+/temperature');
        finally
          vSource.Free;
        end;
      finally
        vMQTT.Free;
      end;
    finally
      vWSClient.Free;
    end;
  finally
    vBuffer.Free;
  end;
end;
```

Véase También

- Descripción General del Puente
- TsgcChartDataSource

TsgcChartDataSource_WebSocket

Unidad: sgcChartsWS_DataSource_WebSocket
Origen de datos de gráfico alimentado por un cliente WebSocket de sgcWebSockets. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Mapping**, **Throttle** y los eventos previos a la adición y de cambio de datos. Consulte Descripción General del Puente para saber cómo funciona el encadenamiento de controladores.
Cada mensaje de texto que recibe el cliente se trata como JSON y se lee a través de **Mapping**. **MessageFilter** es una ruta opcional que debe existir en el mensaje para que se represente en el gráfico; déjelo vacío para representar todos los mensajes.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Active	Boolean	True	Instala o retira el enganche en **Client**.
Client	TsgcWebSocketClient	nil	El cliente WebSocket cuyo `OnMessage` se engancha.
MessageFilter	string	''	Una ruta que debe existir en el mensaje para que se represente en el gráfico. Vacío representa todos los mensajes.

Métodos

Método	Devuelve	Descripción
ProcessMessage(const aText: string)	Boolean	Representa en el gráfico un mensaje como si procediera de **Client**. False cuando el mensaje no es JSON válido o **MessageFilter** no existe en él.

Ejemplo: Representación en el gráfico de un flujo JSON por WebSocket

```
var
  vBuffer: TsgcChartBuffer;
  vSource: TsgcChartDataSource_WebSocket;
  vClient: TsgcWebSocketClient;
begin
  vBuffer := TsgcChartBuffer.Create(4096, cbmOverwrite);
  try
    vClient := TsgcWebSocketClient.Create(nil);
    try
      vClient.URL := 'wss://example.com/feed';

      vSource := TsgcChartDataSource_WebSocket.Create(nil);
      try
        vSource.Buffer := vBuffer;
        vSource.Client := vClient;
        vSource.Mapping.XPath := 'ts';
        vSource.Mapping.TimeFormat := ctfUnixMilliseconds;
```

```
        vSource.Mapping.YPath := 'price';

        vClient.Active := True;
      finally
        vSource.Free;
      end;
    finally
      vClient.Free;
    end;
  finally
    vBuffer.Free;
  end;
end;
```

Véase También

- Descripción General del Puente
- TsgcChartDataSource

TsgcChartDataSource_SSE

Unidad: sgcChartsWS_DataSource_SSE
Origen de datos de gráfico alimentado por el flujo de eventos enviados por el servidor (Server-Sent Events) de un cliente HTTP de sgcWebSockets. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Mapping**, **Throttle** y los eventos previos a la adición y de cambio de datos. Consulte Descripción General del Puente para saber cómo funciona el encadenamiento de controladores.
Los datos de cada evento SSE se leen como JSON a través de **Mapping**. Un evento dividido entre más de un fragmento de flujo se almacena internamente en búfer hasta que está completo.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Active	Boolean	True	Instala o retira el enganche en **Client**.
Client	TsgcHTTP1Client	nil	El cliente HTTP cuyo `OnSSEMessage` se engancha.

Métodos

Método	Devuelve	Descripción
ProcessChunk(const aText: string)	Integer	Representa en el gráfico los eventos completos de un fragmento de flujo, como si procedieran de **Client**. Un evento incompleto espera al siguiente fragmento. Devuelve el número de eventos cuyos datos eran JSON válido.
ResetStream	—	Descarta un evento incompleto almacenado en búfer.

Ejemplo: Representación en el gráfico de un flujo SSE

```
var
  vBuffer: TsgcChartBuffer;
  vSource: TsgcChartDataSource_SSE;
  vClient: TsgcHTTP1Client;
begin
  vBuffer := TsgcChartBuffer.Create(4096, cbmOverwrite);
  try
    vClient := TsgcHTTP1Client.Create(nil);
    try
      vSource := TsgcChartDataSource_SSE.Create(nil);
      try
        vSource.Buffer := vBuffer;
        vSource.Client := vClient;
        vSource.Mapping.XPath := 'ts';
        vSource.Mapping.TimeFormat := ctfUnixSeconds;
        vSource.Mapping.YPath := 'value';

        vClient.GetSSE('https://example.com/stream');
      finally
        vSource.Free;
      end;
    finally
      vClient.Free;
```

```
    end;
  finally
    vBuffer.Free;
  end;
end;
```

Véase También

- Descripción General del Puente
- TsgcChartDataSource

TsgcChartDataSource_Binance

Unidad: sgcChartsWS_DataSource_Binance
Origen de datos de gráfico alimentado por el cliente de la API de Binance de sgcWebSockets. El componente desciende de TsgcChartDataSource y hereda **Buffer**, **OHLCBuffer**, **Throttle** y los eventos previos a la adición y de cambio de datos (no utiliza el **Mapping** heredado, ya que las formas de los mensajes de Binance son fijas por flujo). Consulte Descripción General del Puente para saber cómo funciona el encadenamiento de controladores.
Stream selecciona qué se representa en el gráfico: `cbsKLine` representa velas (en **OHLCBuffer**) construidas a partir de eventos kline/candlestick para **Symbol** e **Interval**; `cbsTicker` y `cbsTrade` representan un valor (en **Buffer**) a partir del flujo de ticker o de operaciones correspondiente para **Symbol**. **ClosedOnly**, cuando se utiliza `cbsKLine`, representa únicamente los klines cuya vela se ha cerrado, en lugar de cada actualización en curso.
Con `cbsTrade`, **OnTrade** informa de cada operación de **Symbol** con su hora, precio y cantidad (el campo `q`) antes de representarla en el gráfico. En la edición financiera, asigne un TsgcChartAggregator a **Aggregator** para construir velas a partir de las operaciones: cada operación pasa entonces a `Aggregator.AddTrade(time, price, quantity)` en lugar de a **Buffer**, de modo que las velas de la serie del agregador incorporan el volumen negociado, las operaciones en sí no se dibujan como una línea y `OnBeforeAppend` no se dispara para ellas. Con **Aggregator** en nil, las operaciones se representan en **Buffer** como antes.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Active	Boolean	True	Instala o retira el enganche en el cliente de **API**.
Aggregator	TsgcChartAggregator	nil	Edición financiera. Con `Stream = cbsTrade`, cada operación se pasa a `Aggregator.AddTrade` con su hora, precio y cantidad en lugar de añadirse a **Buffer**. Se vacía cuando se libera el agregador.
API	TsgcWSAPI_Binance	nil	El componente de la API de Binance cuyos mensajes de cliente se enganchan.
ClosedOnly	Boolean	False	Con `Stream = cbsKLine`, representa únicamente las velas cerradas.
Interval	TsgcWSBinanceChartIntervals	bci1m	Intervalo de kline, utilizado cuando `Stream = cbsKLine`.
Stream	TsgcChartBinanceStream	cbsKLine	`cbsKLine`, `cbsTicker` o `cbsTrade`.
Symbol	string	''	El símbolo de Binance que se va a representar en el gráfico, por ejemplo `BTCUSDT`.

Métodos

Método	Devuelve	Descripción
Subscribe	Integer	Se suscribe a **Stream** para **Symbol** (y **Interval** para los klines). Devuelve el id que devuelve la API.

Unsubscribe	Integer	Cancela la suscripción al mismo flujo. Devuelve el id que devuelve la API.
ProcessMessage(const aText: string)	Boolean	Representa en el gráfico un mensaje de Binance como si procediera de **API.Client**. False cuando no es un mensaje de **Stream**/**Symbol**/**Interval**, o no se representó nada.

Eventos

Evento	Descripción
OnTrade(Sender: TObject; aTime: TDateTime; aPrice, aQty: Double)	Público. Con `Stream = cbsTrade` , se dispara para cada operación de **Symbol** mientras el origen está activo, antes de que la operación se añada o se agregue: aTime es la hora de la operación (UTC), aPrice el precio y aQty la cantidad (0 cuando el mensaje no tiene `q`).

Ejemplo: Representación en el gráfico de klines de Binance en vivo

```
var
  vOHLC: TsgcChartOHLCBuffer;
  vSource: TsgcChartDataSource_Binance;
  vAPI: TsgcWSAPI_Binance;
begin
  vOHLC := TsgcChartOHLCBuffer.Create(2048, cbmOverwrite);
  try
    vAPI := TsgcWSAPI_Binance.Create(nil);
    try
      vSource := TsgcChartDataSource_Binance.Create(nil);
      try
        vSource.OHLCBuffer := vOHLC;
        vSource.API := vAPI;
        vSource.Stream := cbsKLine;
        vSource.Symbol := 'BTCUSDT';
        vSource.Interval := bci1m;
        vSource.ClosedOnly := False;

        vSource.Subscribe;
      finally
        vSource.Free;
      end;
    finally
      vAPI.Free;
    end;
  finally
    vOHLC.Free;
  end;
end;
```

Véase También

- Descripción General del Puente
- TsgcChartDataSource

TsgcChartSeries_Indicator

Unidad: sgcCharts_Series_Financial
Solo en las ediciones Registered y Trial (excluida de la edición Community; véase Ediciones).
TsgcChartSeries_Indicator es la base abstracta de cada serie de indicador financiero: un descendiente de TsgcChartSeries que lee las barras de otra serie, su **Source**, una serie de velas u OHLC o una serie X/Y cuya Y hace las veces de apertura, máximo, mínimo y cierre sin volumen, y mantiene sus propias salidas calculadas en matrices internas en lugar de dibujar los datos de origen. Los indicadores concretos, TsgcChartSeries_SMA, TsgcChartSeries_EMA, TsgcChartSeries_WMA, TsgcChartSeries_Bollinger y TsgcChartSeries_VWAP, se dibujan en el mismo eje que el precio, por lo que normalmente se añaden sobre una serie de velas. La familia de osciladores (TsgcChartSeries_Oscillator y sus descendientes RSI y Stochastic), MACD, ATR y TsgcChartSeries_Volume también descienden de TsgcChartSeries_Indicator, pero se documentan en sus propios apartados porque normalmente se dibujan en un panel aparte, debajo del precio.

Propiedades

Propiedad	Tipo	Descripción
Source	TsgcChartSeries	La serie que este indicador sigue: una serie de velas / OHLC, o una serie X/Y (su Y se utiliza como apertura, máximo, mínimo y cierre, sin volumen). Cambiarla vacía las salidas y fuerza un cálculo completo en el siguiente **Sync**.
LineWidth	Double	Píxeles de dispositivo a escala 1. Predeterminado 1.5.
OutputCount	Integer (read-only)	Número de líneas de salida que calcula el indicador por barra, 1 para un indicador de una sola línea como SMA, más para uno de varias líneas como Bollinger.
PointCount	Integer (read-only)	Número de barras procesadas, siendo 0 la más antigua.
XValues[aIndex: Integer]	Double (solo lectura)	El X (normalmente la hora de la barra) de la barra procesada en aIndex.
Values[aLine, aIndex: Integer]	Double (solo lectura)	Salida aLine (basada en 0, hasta **OutputCount** - 1) de la barra procesada en aIndex. NaN durante el período de calentamiento del indicador, cuando todavía no se han visto suficientes barras para producir un valor.

Métodos

Método	Devuelve	Descripción
Sync		Pone las salidas al día con **Source**. Véase Sincronización con el origen.
Recalculate		Fuerza un cálculo completo en el siguiente **Sync**, por ejemplo después de reescribir en el sitio barras más antiguas de **Source**.
GetDataRange(var aXMin, aXMax, aYMin, aYMax: Double)	Boolean	Llama primero a **Sync**, y luego devuelve los extremos X e Y de las salidas; False cuando no hay ninguna.

Draw(aCanvas: TsgcUICanvas; aXAxis, aYAxis: TsgcChartAxis; const aPlot: TsgcUIRect; aScale: Double)		Pinta cada salida como una línea de **EffectiveColor**. Un descendiente como Bollinger lo sobrescribe para dibujar bandas o barras en su lugar.
DrawLegendSymbol(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Pinta el símbolo de la leyenda para este indicador.

Sincronización con el origen

Un indicador nunca copia los datos de **Source**; lee sus barras y conserva únicamente sus propias salidas. **Sync** es lo que pone esas salidas al día con **Source** de nuevo, y se llama automáticamente desde **GetDataRange** (por lo tanto en cada renderizado) y desde la prueba de impacto, además de estar disponible para llamarlo directamente. Lo que hace depende de lo que haya cambiado desde la última llamada:

- Sin salidas todavía, un cambio de parámetro (como establecer **Period**), o un origen vaciado o recargado (menos barras que las ya procesadas, o la barra procesada más reciente tiene ahora una X diferente): un cálculo completo, que reinicia el indicador y lo alimenta de nuevo con cada barra, con el mismo resultado que calcularlo desde cero.
- Barras añadidas a **Source** desde el último Sync: solo se introducen las barras nuevas.
- La barra procesada más reciente cambió en el sitio, el caso de la vela en vivo (véase **TsgcChartSeries_OHLCBase.UpdateLastOHLC**): solo se recalcula esa última salida.
- Barras descartadas al principio de un búfer circular lleno (**BufferMode** = `cbmOverwrite` en el origen): las salidas de las barras descartadas se desechan y el cálculo continúa desde donde estaba, de modo que las salidas restantes conservan el historial con el que se calcularon.

Sync detecta qué caso se aplica comparando el número de barras de **Source**, la X de su barra más antigua y de la barra procesada más reciente, y los valores OHLCV de esa barra más reciente. Llame a **Recalculate** después de reescribir en el sitio barras más antiguas de **Source**, ya que Sync no puede detectar ese caso por sí solo. Las salidas todavía en su período de calentamiento, antes de que el indicador haya visto suficientes barras para producir un valor, se leen como NaN a través de **Values** y no se dibujan. Todo se ejecuta en el hilo principal; el búfer de origen se lee bajo su propio bloqueo.

TsgcChartSeries_SMA

Media móvil simple del cierre. Una línea de salida, índice 0.

Propiedad	Tipo	Descripción
Period	Integer	Número de barras promediadas. Predeterminado 20.

TsgcChartSeries_EMA

Media móvil exponencial del cierre, que pondera las barras recientes más que una media simple. Una línea de salida, índice 0.

Propiedad	Tipo	Descripción
Period	Integer	Número de barras de las que se deriva el factor de suavizado. Predeterminado 20.

TsgcChartSeries_WMA

Media móvil ponderada linealmente del cierre, que pondera más la barra más reciente del período y menos la más antigua. Una línea de salida, índice 0.

Propiedad	Tipo	Descripción

Period	Integer	Número de barras ponderadas. Predeterminado 20.

TsgcChartSeries_Bollinger

Bandas de Bollinger: una media móvil del cierre con una banda superior e inferior a un número elegido de desviaciones estándar de ella. Como ambas bandas siguen la misma escala de precio que el origen, Bollinger se dibuja sobre el precio como las medias móviles, no en su propio panel. Tres líneas de salida: 0 es la banda media, 1 la banda superior, 2 la banda inferior.

Propiedad	Tipo	Descripción
Period	Integer	Número de barras sobre las que se calculan la banda media y la desviación estándar. Predeterminado 20.
Deviations	Double	Número de desviaciones estándar a las que se sitúan las bandas superior e inferior respecto a la banda media. Predeterminado 2.
BandFill	Boolean	Rellena el área entre la banda superior y la inferior. Predeterminado True.

TsgcChartSeries_VWAP

Precio medio ponderado por volumen. Necesita un origen que lleve volumen, una serie OHLC o de velas alimentada mediante **AddOHLC** con un volumen distinto de cero; un origen X/Y no tiene volumen y promedia como si cada barra pesara lo mismo. La X del origen se toma como la hora de la barra. Una línea de salida, índice 0.

Propiedad	Tipo	Descripción
SessionStart	TDateTime	Fracción del día en la que comienza una sesión, 0 (el valor predeterminado) significa medianoche; el cálculo se reinicia en cada límite de sesión.

Ejemplo: SMA, EMA y bandas de Bollinger sobre una serie de velas

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Financial,
  sgcCharts_Axis, sgcCharts_Types, sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vAxisX, vAxisY: TsgcChartAxis;
  vPrice: TsgcChartSeries_Candlestick;
  vSMA: TsgcChartSeries_SMA;
  vBollinger: TsgcChartSeries_Bollinger;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
  vClose, vOpen: Double;
begin
  vChart := TsgcChart.Create(nil);
  try
    vAxisX := vChart.Axes.Add;
    vAxisX.Name := 'bottom';
    vAxisX.Position := capBottom;
    vAxisY := vChart.Axes.Add;
    vAxisY.Name := 'left';
    vAxisY.Position := capLeft;
```

```
    vPrice := TsgcChartSeries_Candlestick.Create(nil);
    vSMA := TsgcChartSeries_SMA.Create(nil);
    vBollinger := TsgcChartSeries_Bollinger.Create(nil);
    try
      vPrice.Chart := vChart;
      vPrice.AxisX := 'bottom';
      vPrice.AxisY := 'left';

      vClose := 100;
      for i := 0 to 79 do
      begin
        vOpen := vClose;
        vClose := vOpen + Sin(i / 6) * 2 + (Random - 0.5);
        vPrice.AddOHLC(i, vOpen, Max(vOpen, vClose) + 1,
          Min(vOpen, vClose) - 1, vClose, 1000 + Random(500));
      end;

      vSMA.Chart := vChart;
      vSMA.AxisX := 'bottom';
      vSMA.AxisY := 'left';
      vSMA.Source := vPrice;
      vSMA.Period := 20;

      vBollinger.Chart := vChart;
      vBollinger.AxisX := 'bottom';
      vBollinger.AxisY := 'left';
      vBollinger.Source := vPrice;
      vBollinger.Period := 20;
      vBollinger.Deviations := 2;

      vSMA.Sync;
      vBollinger.Sync;
      Writeln('SMA last value: ',
        vSMA.Values[0, vSMA.PointCount - 1]: 0: 4);
      Writeln('Bollinger upper/middle/lower: ',
        vBollinger.Values[1, vBollinger.PointCount - 1]: 0: 4, ' / ',
        vBollinger.Values[0, vBollinger.PointCount - 1]: 0: 4, ' / ',
        vBollinger.Values[2, vBollinger.PointCount - 1]: 0: 4);

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
        vRaster.SaveToPNGFile('indicator.png');
      finally
        vRaster.Free;
      end;
    finally
      vBollinger.Free;
      vSMA.Free;
      vPrice.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChartSeries_Oscillator
- TsgcChartSeries_Volume
- TsgcChartAggregator
- TsgcChartDrawing
- Panes
- TsgcChartSeries_Candlestick

- TsgcChartSeries
- Ediciones

TsgcChartSeries_Oscillator

Unidad: sgcCharts_Series_Financial
Solo en las ediciones Registered y Trial (excluida de la edición Community; véase Ediciones).
TsgcChartSeries_Oscillator es la base de los indicadores de estilo 0..100: un descendiente de TsgcChartSeries_Indicator que fija sus extremos Y a 0..100 independientemente de los datos y puede dibujar un par de líneas guía discontinuas de sobrecompra / sobreventa. TsgcChartSeries_RSI y TsgcChartSeries_Stochastic descienden de ella. **TsgcChartSeries_MACD** y **TsgcChartSeries_ATR** se documentan más abajo en esta misma página porque normalmente también se dibujan en su propio panel, pero descienden directamente de TsgcChartSeries_Indicator, no de TsgcChartSeries_Oscillator: su salida es un valor derivado del precio sin un techo natural de 0..100, por lo que mantienen un rango Y flotante en lugar de uno fijo. Los cuatro indicadores normalmente se colocan en un panel propio debajo del panel de precio, ya que su escala no coincide con el precio.

Propiedades

Propiedad	Tipo	Descripción
GuideLines	Boolean	Dibuja las líneas guía de sobrecompra y sobreventa como líneas horizontales discontinuas. Predeterminado True.

Overbought y Oversold no se publican en la propia clase base; cada oscilador concreto las publica con su propio nivel predeterminado, ya que RSI y Stochastic no coinciden en dónde se sitúan esos niveles. Véase TsgcChartSeries_RSI y TsgcChartSeries_Stochastic más abajo.
Cada miembro de TsgcChartSeries_Indicator, incluidos **Source**, las propiedades de estilo **Period** de las clases concretas, **Sync**, **PointCount** y **Values**, sigue aplicándose aquí; véase Sincronización con el origen para saber cómo un indicador se mantiene al día con su serie **Source**.

TsgcChartSeries_RSI

Índice de fuerza relativa del cierre. Una línea de salida, índice 0. Las líneas guía son 30 y 70 de forma predeterminada.

Propiedad	Tipo	Descripción
Period	Integer	Número de barras sobre las que se calculan la ganancia y la pérdida medias. Predeterminado 14.
Overbought	Double	Nivel de la línea guía. Predeterminado 70.
Oversold	Double	Nivel de la línea guía. Predeterminado 30.

TsgcChartSeries_Stochastic

Oscilador estocástico. Dos líneas de salida: 0 es %K, 1 es %D (una media móvil de %K). Las líneas guía son 20 y 80 de forma predeterminada.

Propiedad	Tipo	Descripción
KPeriod	Integer	Período retrospectivo para %K. Predeterminado 14.

KSmoothing	Integer	Período de suavizado aplicado a %K. Predeterminado 3.
DPeriod	Integer	Período de la media móvil de %K que produce %D. Predeterminado 3.
DColor	TsgcUIColor	Color de la línea %D; %K se dibuja en **EffectiveColor** como un indicador de una sola línea. Predeterminado a un color de acento naranja.
Overbought	Double	Nivel de la línea guía. Predeterminado 80.
Oversold	Double	Nivel de la línea guía. Predeterminado 20.

TsgcChartSeries_MACD

Convergencia / divergencia de medias móviles del cierre. Desciende directamente de TsgcChartSeries_Indicator: su escala sigue las diferencias de precio a partir de las cuales se construye, por lo que no tiene un rango fijo de 0..100 ni líneas guía de sobrecompra / sobreventa. Sí que sobrescribe el rango Y para que la línea base cero que atraviesa el histograma sea siempre visible. Tres líneas de salida: 0 es la línea MACD, 1 la línea de señal, 2 el histograma (línea MACD menos línea de señal).

Propiedad	Tipo	Descripción
FastPeriod	Integer	Período de la EMA rápida. Predeterminado 12.
SlowPeriod	Integer	Período de la EMA lenta. Predeterminado 26.
SignalPeriod	Integer	Período de la EMA de la línea MACD que produce la línea de señal. Predeterminado 9.
SignalColor	TsgcUIColor	Color de la línea de señal; la línea MACD utiliza **EffectiveColor** y el histograma se dibuja como barras de subida / bajada. Predeterminado a un color de acento naranja.

TsgcChartSeries_ATR

Rango medio verdadero, una medida de volatilidad. Desciende directamente de TsgcChartSeries_Indicator por el mismo motivo que MACD: su valor es un rango de precio positivo sin límite, no una proporción 0..100, por lo que mantiene un rango Y flotante y no tiene líneas guía. Una línea de salida, índice 0.

Propiedad	Tipo	Descripción
Period	Integer	Número de barras promediadas. Predeterminado 14.

Ejemplo: RSI y MACD en sus propios paneles

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Financial,
  sgcCharts_Axis, sgcCharts_Types, sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vAxisX, vAxisY: TsgcChartAxis;
  vPrice: TsgcChartSeries_Candlestick;
  vRSI: TsgcChartSeries_RSI;
  vMACD: TsgcChartSeries_MACD;
```

```
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
  vClose, vOpen: Double;
begin
  vChart := TsgcChart.Create(nil);
  try
    vAxisX := vChart.Axes.Add;
    vAxisX.Name := 'bottom';
    vAxisX.Position := capBottom;
    vAxisY := vChart.Axes.Add;
    vAxisY.Name := 'left';
    vAxisY.Position := capLeft;

    vPrice := TsgcChartSeries_Candlestick.Create(nil);
    vRSI := TsgcChartSeries_RSI.Create(nil);
    vMACD := TsgcChartSeries_MACD.Create(nil);
    try
      vPrice.Chart := vChart;
      vPrice.AxisX := 'bottom';
      vPrice.AxisY := 'left';

      vClose := 100;
      for i := 0 to 79 do
      begin
        vOpen := vClose;
        vClose := vOpen + Sin(i / 6) * 2 + (Random - 0.5);
        vPrice.AddOHLC(i, vOpen, Max(vOpen, vClose) + 1,
          Min(vOpen, vClose) - 1, vClose, 1000 + Random(500));
      end;

      vRSI.Chart := vChart;
      vRSI.AxisX := 'bottom';
      vRSI.AxisY := 'left';
      vRSI.Source := vPrice;
      vRSI.Period := 14;

      vMACD.Chart := vChart;
      vMACD.AxisX := 'bottom';
      vMACD.AxisY := 'left';
      vMACD.Source := vPrice;
      vMACD.FastPeriod := 12;
      vMACD.SlowPeriod := 26;
      vMACD.SignalPeriod := 9;

      vRSI.Sync;
      vMACD.Sync;
      Writeln('RSI last value: ', vRSI.Values[0, vRSI.PointCount - 1]: 0: 4);
      Writeln('MACD line/signal/histogram: ',
        vMACD.Values[0, vMACD.PointCount - 1]: 0: 4, ' / ',
        vMACD.Values[1, vMACD.PointCount - 1]: 0: 4, ' / ',
        vMACD.Values[2, vMACD.PointCount - 1]: 0: 4);

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
        vRaster.SaveToPNGFile('oscillator.png');
      finally
        vRaster.Free;
      end;
    finally
      vMACD.Free;
      vRSI.Free;
      vPrice.Free;
    end;
  finally
    vChart.Free;
```

```
  end;
end;
```

Véase También

- TsgcChartSeries_Indicator
- TsgcChartSeries_Volume
- Panes
- TsgcChartSeries_Candlestick
- TsgcChartSeries
- Ediciones

TsgcChartSeries_Volume

Unidad: sgcCharts_Series_Financial
Solo ediciones Registered y Trial (excluida de la edición Community; véase Ediciones).
TsgcChartSeries_Volume dibuja las barras de volumen de un **Source** OHLC, coloreadas según si esa barra cerró al alza o a la baja. Desciende directamente de TsgcChartSeries_Indicator (no de TsgcChartSeries_Oscillator): el volumen no tiene un techo fijo, así que mantiene un rango Y flotante con su línea base fijada en cero en lugar de una escala 0..100. Calcula dos salidas por barra, 0 es el volumen y 1 es la dirección (1 para cierre >= apertura, 0 en caso contrario), pero solo la salida 0 cuenta para los extremos de Y, de modo que el indicador de dirección nunca estira el eje. Como RSI, Stochastic, MACD y ATR, normalmente se coloca en su propio panel debajo del precio, ya que su escala no coincide con la del precio. Véase Sincronización con la fuente para saber cómo mantiene sus salidas alineadas con **Source**, se aplican aquí las mismas reglas.

Propiedades

Propiedad	Tipo	Descripción
UpColor	TsgcUIColor	Color de una barra cuyo cierre es mayor o igual que su apertura. Toma como valor predeterminado el color alcista del gráfico con opacidad reducida.
DownColor	TsgcUIColor	Color de una barra cuyo cierre es menor que su apertura. Toma como valor predeterminado el color bajista del gráfico con opacidad reducida.
BarWidthPercent	Integer	Proporción del hueco X ocupada por una barra, 1 a 100. Valor predeterminado 70.

También se aplica todo miembro de TsgcChartSeries_Indicator: **Source**, **Sync**, **Recalculate**, **PointCount**, **XValues** y **Values**. Lea el volumen de la barra procesada en aIndex con `Values[0, aIndex]` y su indicador de alza/baja con `Values[1, aIndex]`.

Métodos

Método	Devuelve	Descripción
Draw(aCanvas: TsgcUICanvas; aXAxis, aYAxis: TsgcChartAxis; const aPlot: TsgcUIRect; aScale: Double)		Sobrescrito para pintar barras de volumen, las barras alcistas en **UpColor** y las bajistas en **DownColor**, en lugar de la línea que dibuja de forma predeterminada TsgcChartSeries_Indicator.
DrawLegendSymbol(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Sobrescrito para que coincida con el símbolo de leyenda de barra en lugar del símbolo de línea.

Ejemplo: barras de volumen bajo una serie de velas

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Financial,
  sgcCharts_Axis, sgcCharts_Types, sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
```

```
  vChart: TsgcChart;
  vAxisX, vAxisY: TsgcChartAxis;
  vPrice: TsgcChartSeries_Candlestick;
  vVolume: TsgcChartSeries_Volume;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
  vClose, vOpen, vVol: Double;
begin
  vChart := TsgcChart.Create(nil);
  try
    vAxisX := vChart.Axes.Add;
    vAxisX.Name := 'bottom';
    vAxisX.Position := capBottom;
    vAxisY := vChart.Axes.Add;
    vAxisY.Name := 'left';
    vAxisY.Position := capLeft;

    vPrice := TsgcChartSeries_Candlestick.Create(nil);
    vVolume := TsgcChartSeries_Volume.Create(nil);
    try
      vPrice.Chart := vChart;
      vPrice.AxisX := 'bottom';
      vPrice.AxisY := 'left';

      vClose := 100;
      for i := 0 to 79 do
      begin
        vOpen := vClose;
        vClose := vOpen + Sin(i / 6) * 2 + (Random - 0.5);
        vVol := 1000 + Random(500);
        vPrice.AddOHLC(i, vOpen, Max(vOpen, vClose) + 1,
          Min(vOpen, vClose) - 1, vClose, vVol);
      end;

      vVolume.Chart := vChart;
      vVolume.AxisX := 'bottom';
      vVolume.AxisY := 'left';
      vVolume.Source := vPrice;
      vVolume.BarWidthPercent := 70;

      vVolume.Sync;
      Writeln('last volume: ',
        vVolume.Values[0, vVolume.PointCount - 1]: 0: 2);
      Writeln('last bar up: ',
        vVolume.Values[1, vVolume.PointCount - 1] <> 0);

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
        vRaster.SaveToPNGFile('volume.png');
      finally
        vRaster.Free;
      end;
    finally
      vVolume.Free;
      vPrice.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase también

- TsgcChartSeries_Indicator

- TsgcChartSeries_Oscillator
- Panes
- TsgcChartSeries_Candlestick
- TsgcChartSeries
- Ediciones

Panes

Unidad: sgcCharts_Chart
Solo en las ediciones Registered y Trial (excluida de la edición Community; véase Ediciones).
Un TsgcChart normalmente tiene un área de trazado. Cuando **Panes** contiene una o más entradas, esa área de trazado se divide en bandas horizontales apiladas de arriba abajo, cada una con su propio eje vertical y sus propias series, mientras que cada panel sigue compartiendo los ejes horizontales del gráfico. Este es el diseño que necesita un gráfico de precios con RSI, MACD o volumen debajo: las velas de precio en un panel, un oscilador en otro debajo, ambos desplazándose y con zoom juntos porque comparten el mismo eje inferior. **TsgcChartPane** es una banda de la colección, **TsgcChartPanes** la colección en sí, y un eje se une a un panel mediante **TsgcChartAxis.Pane**.

Propiedades de TsgcChartPane

Propiedad	Tipo	Descripción
Name	string	Se busca mediante **TsgcChartAxis.Pane** y mediante **TsgcChartPanes.Find** (sin distinguir mayúsculas y minúsculas).
HeightPercent	Double	Peso del panel: los paneles visibles se reparten la altura del trazado en proporción a él, no como un porcentaje literal que deba sumar 100. Predeterminado 100.
Visible	Boolean	Un panel oculto no ocupa altura de trazado y sus ejes y series no se dibujan. Predeterminado True.

Métodos de TsgcChartPanes

Método	Devuelve	Descripción
Add	TsgcChartPane	Añade un panel al final de la pila.
Find(const aName: string)	TsgcChartPane	Primer panel con este Name (sin distinguir mayúsculas y minúsculas), nil cuando no hay ninguno.

Propiedades y métodos de TsgcChart

Miembro	Tipo	Descripción
Panes	TsgcChartPanes (property)	Vacío (el valor predeterminado) significa un área de trazado; una colección no vacía divide el trazado entre los paneles visibles, de arriba abajo en el orden de la colección.
HasPanes	Boolean (método)	True cuando el último **Render** dividió el trazado en paneles.
PaneRect(aIndex: Integer)	TsgcUIRect (method)	Rectángulo del panel en este índice de la colección en el último **Render**, vacío cuando el panel está oculto.
AxisPlotRect(aAxis: TsgcChartAxis)	TsgcUIRect (method)	Rectángulo de trazado utilizado por aAxis en el último **Render**: el panel de un eje vertical cuando el gráfico tiene paneles (vacío cuando su panel está oculto), o si no, todo el **PlotRect**.

SeriesPlotRect(aSeries: TsgcChartSeries)	TsgcUIRect (method)	Rectángulo de trazado del panel de aSeries, el panel de su eje vertical.
SplitterAt(X, Y: Double)	Integer (método)	Índice del panel por encima del divisor bajo X, Y (píxeles del gráfico, el hueco entre dos paneles visibles), -1 cuando el punto no está sobre un divisor.
MoveSplitter(aIndex: Integer; Y: Double)	(método)	Mueve el divisor situado debajo del panel en aIndex hasta Y (píxeles del gráfico): cambia el **HeightPercent** de ese panel y del siguiente visible, manteniendo igual su suma.

Enlace de un eje y una serie a un panel

TsgcChartAxis.Pane (una cadena, vacía de forma predeterminada) nombra el panel al que pertenece un eje vertical; un **Pane** vacío se resuelve al primer panel visible. Un eje horizontal ignora **Pane** por completo, ya que cada panel comparte los mismos ejes horizontales. Una serie no tiene una propiedad **Pane** propia: se dibuja en el panel de su **YAxis**, de modo que colocar una serie en un panel es enteramente cuestión de qué eje vertical nombra su **AxisY**, y del propio **Pane** de ese eje. Un gráfico con un panel de precio y un panel de RSI debajo suele tener tres ejes, un eje inferior compartido y un eje izquierdo por panel, cada eje izquierdo con su **Pane** establecido en el **Name** del panel correspondiente.

Arrastre del divisor

Los paneles no se redimensionan automáticamente: un anfitrión que permite al usuario arrastrar el hueco entre dos paneles llama a **SplitterAt** al pulsar el ratón para averiguar si el puntero está sobre un divisor (y qué panel está por encima de él), y luego llama a **MoveSplitter** con ese índice de panel en cada movimiento del ratón durante el arrastre, pasando la nueva Y en píxeles del gráfico. **MoveSplitter** mantiene constante la suma de **HeightPercent** de los dos paneles a cada lado del divisor, de modo que los paneles más abajo en la pila no se ven afectados. Así es exactamente como TsgcChartInteraction gestiona el arrastre del divisor a partir de los eventos de ratón de VCL y FMX; el código que renderiza sin una vista y nunca recibe entrada de ratón no tiene motivo para llamar a ninguno de los dos métodos.

Ejemplo: Un panel de precio y un panel de RSI que comparten el eje inferior

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Financial,
  sgcCharts_Axis, sgcCharts_Types, sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vPrice: TsgcChartSeries_Candlestick;
  vRSI: TsgcChartSeries_RSI;
  vAxisBottom, vAxisPrice, vAxisRSI: TsgcChartAxis;
  vRaster: TsgcUICanvas_Raster;
  i, vSplitter: Integer;
  vClose, vOpen: Double;
  vPricePane, vRSIPane: TsgcUIRect;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Panes.Add.Name := 'price';
    vChart.Panes[0].HeightPercent := 70;
    vChart.Panes.Add.Name := 'rsi';
    vChart.Panes[1].HeightPercent := 30;

    vAxisBottom := vChart.Axes.Add;
    vAxisBottom.Name := 'bottom';
    vAxisBottom.Position := capBottom;
```

```
    vAxisPrice := vChart.Axes.Add;
    vAxisPrice.Name := 'price';
    vAxisPrice.Position := capLeft;
    vAxisPrice.Pane := 'price';

    vAxisRSI := vChart.Axes.Add;
    vAxisRSI.Name := 'rsi';
    vAxisRSI.Position := capLeft;
    vAxisRSI.Pane := 'rsi';

    vPrice := TsgcChartSeries_Candlestick.Create(nil);
    vRSI := TsgcChartSeries_RSI.Create(nil);
    try
      vPrice.Chart := vChart;
      vPrice.AxisX := 'bottom';
      vPrice.AxisY := 'price';

      vClose := 100;
      for i := 0 to 79 do
      begin
        vOpen := vClose;
        vClose := vOpen + Sin(i / 6) * 2 + (Random - 0.5);
        vPrice.AddOHLC(i, vOpen, Max(vOpen, vClose) + 1,
          Min(vOpen, vClose) - 1, vClose, 1000 + Random(500));
      end;

      vRSI.Chart := vChart;
      vRSI.AxisX := 'bottom';
      vRSI.AxisY := 'rsi';
      vRSI.Source := vPrice;
      vRSI.Period := 14;

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
        Writeln('has panes: ', vChart.HasPanes);

        vPricePane := vChart.PaneRect(0);
        vRSIPane := vChart.PaneRect(1);
        Writeln('price pane bottom: ', vPricePane.Bottom: 0: 2);
        Writeln('rsi pane top: ', vRSIPane.Top: 0: 2);

        vSplitter := vChart.SplitterAt((vPricePane.Left + vPricePane.Right) / 2,
          vPricePane.Bottom);
        Writeln('splitter above pane: ', vSplitter);
        if vSplitter >= 0 then
        begin
          vChart.MoveSplitter(vSplitter, vPricePane.Bottom + 20);
          vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
          Writeln('price pane bottom after drag: ',
            vChart.PaneRect(0).Bottom: 0: 2);
        end;

        vRaster.SaveToPNGFile('panes.png');
      finally
        vRaster.Free;
      end;
    finally
      vRSI.Free;
      vPrice.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChartSeries_Indicator
- TsgcChartSeries_Oscillator
- TsgcChartSeries_Volume
- TsgcChartInteraction
- Ejes
- TsgcChart
- Ediciones

TsgcChartAggregator

Unidad: sgcCharts_Aggregator
Solo en las ediciones Registered y Trial (excluida de la edición Community; véase Ediciones).
TsgcChartAggregator es un `TComponent` que construye las velas de **Series**, una serie de velas u OHLC, a partir de un flujo de operaciones: llame a **AddTrade(aTime, aPrice, aQty)** por cada operación, siendo aTime un `TDateTime` UTC. Cada vela cubre un intervalo de **Interval** y su X es el inicio de ese intervalo (la hora de apertura, la misma convención que un kline de Binance). Una operación dentro del intervalo de la vela en formación actualiza su máximo, mínimo, cierre y volumen en el sitio; una operación en un intervalo posterior cierra esa vela y añade una nueva abierta al inicio de su propio intervalo, con **FillGaps** insertando opcionalmente una vela plana por cada intervalo vacío intermedio; una operación más antigua que la vela en formación se ignora y se cuenta en **DroppedCount**. La primera operación tras la creación, o tras **Reset**, continúa la vela más reciente de **Series** cuando esa vela cae en el mismo intervalo (el historial ya se cargó antes de que empezaran las operaciones en vivo), y se descarta cuando la vela más reciente pertenece a un intervalo posterior.
Los intervalos se calculan en milisegundos completos, alineados en UTC: los segundos, minutos y horas se alinean con el reloj, los días a las 00:00, las semanas el lunes a las 00:00 y los meses el primer día del mes natural. **SessionAlign** desplaza cada límite un número de minutos, para sesiones que no comienzan a las 00:00 UTC.

Propiedades

Propiedad	Tipo	Descripción
Series	TsgcChartSeries_OHLCBase	La serie de velas que recibe las velas. Cambiarla, o que la serie se libere, olvida la vela en formación de la misma manera que lo hace **Reset**.
Interval	TsgcChartAggregatorInterval	Tamaño del intervalo: `cai1s`, `cai5s`, `cai15s`, `cai30s`, `cai1m`, `cai3m`, `cai5m`, `cai15m`, `cai30m`, `cai1h`, `cai2h`, `cai4h`, `cai6h`, `cai12h`, `cai1d`, `cai1w` o `cai1Month`. Predeterminado `cai1m`. Cambiarlo olvida la vela en formación.
SessionAlign	Integer	Minutos añadidos a cada límite UTC, 0 (el valor predeterminado) significa límites UTC simples. Cambiarlo olvida la vela en formación.
FillGaps	Boolean	Inserta una vela plana (open = high = low = close = el cierre anterior, volumen 0) por cada intervalo vacío entre dos operaciones. Predeterminado False.
DroppedCount	Integer (read-only)	Número de operaciones ignoradas por ser más antiguas que el intervalo de la vela en formación.

Métodos

Método	Devuelve	Descripción
AddTrade(aTime: TDateTime; aPrice, aQty: Double)		Alimenta una operación. Actualiza **Series** de la misma manera que lo harían **AddOHLC** y **UpdateLastOHLC**, por lo que debe llamarse desde el mismo hilo desde el que se llama a esos métodos.
Reset		Olvida la vela en formación y vacía **DroppedCount**. La propia **Series** no cambia.

BucketStart(aTime: TDate-Time)	TDate-Time	Inicio del intervalo en el que cae aTime, respetando **Interval** y **SessionAlign**.

Alimentación del agregador

Las operaciones de Binance se agregan mediante un TsgcChartDataSource_Binance en cuatro pasos: colocar una serie de velas u OHLC en el gráfico (un TsgcChartSeries_Volume en su propio panel puede mostrar el volumen de las velas); asignar esa serie a **Series** y elegir **Interval**; establecer el `Stream` del origen de datos en `cbsTrade`; asignar el agregador a su propiedad `Aggregator`. Cada operación llega entonces a **AddTrade** con su hora, precio y cantidad, de modo que las velas incorporan el volumen negociado, y las operaciones no se dibujan como una línea. Cualquier otra fuente llama a **AddTrade** por sí misma, desde el hilo que produzca las operaciones, siempre que sea el hilo desde el que se actualiza **Series** normalmente.

Ejemplo: Agregación de operaciones en velas de un minuto

```
uses
  SysUtils, DateUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Financial,
  sgcCharts_Aggregator, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vPrice: TsgcChartSeries_Candlestick;
  vAggregator: TsgcChartAggregator;
  vAxisBottom, vAxisLeft: TsgcChartAxis;
  vRaster: TsgcUICanvas_Raster;
  vBase: TDateTime;
begin
  vChart := TsgcChart.Create(nil);
  try
    { TsgcChart already carries two default axes ('X' and 'Y'); adding named
      ones and keeping a reference to each avoids indexing into Axes, whose
      order includes those defaults }
    vAxisBottom := vChart.Axes.Add;
    vAxisBottom.Name := 'bottom';
    vAxisBottom.Position := capBottom;
    vAxisBottom.AxisType := catDateTime;

    vAxisLeft := vChart.Axes.Add;
    vAxisLeft.Name := 'left';
    vAxisLeft.Position := capLeft;

    vPrice := TsgcChartSeries_Candlestick.Create(nil);
    vAggregator := TsgcChartAggregator.Create(nil);
    try
      vPrice.Chart := vChart;
      vPrice.AxisX := 'bottom';
      vPrice.AxisY := 'left';

      vAggregator.Series := vPrice;
      vAggregator.Interval := cai1m;
      vAggregator.FillGaps := True;

      vBase := EncodeDateTime(2026, 9, 28, 9, 0, 0, 0);
      vAggregator.AddTrade(vBase, 100, 1);
      vAggregator.AddTrade(vBase + 10 / SecsPerDay, 100.5, 2);
      vAggregator.AddTrade(vBase + 40 / SecsPerDay, 99.8, 1);
      // two minutes after the base trade: seals the forming candle and,
      // with FillGaps, inserts the flat candle for the empty minute between
      vAggregator.AddTrade(vBase + 2 * 60 / SecsPerDay, 101, 3);
      // older than the forming candle: dropped, counted in DroppedCount
      vAggregator.AddTrade(vBase, 90, 1);
```

```
      Writeln('bucket start of the base trade: ',
        DateTimeToStr(vAggregator.BucketStart(vBase)));
      Writeln('candles built: ', vPrice.OHLCBuffer.Count);
      Writeln('dropped trades: ', vAggregator.DroppedCount);

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));
        vRaster.SaveToPNGFile('aggregator.png');
      finally
        vRaster.Free;
      end;
    finally
      vAggregator.Free;
      vPrice.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChartSeries_Indicator
- TsgcChartDrawing
- Panes
- Descripción General del Puente
- TsgcChartDataSource_Binance
- TsgcChartSeries_Candlestick
- Ediciones

TsgcChartDrawing

Unidad: sgcCharts_Drawing
Solo en las ediciones Registered y Trial (excluida de la edición Community; véase Ediciones).
TsgcChart.Drawings contiene elementos **TsgcChartDrawing**: líneas de tendencia, líneas horizontales y verticales, rectángulos y retrocesos de Fibonacci, anclados en valores de datos (**X1**, **Y1**, **X2**, **Y2**) de los ejes indicados por **AxisX** y **AxisY**, de modo que un dibujo sigue el zoom, el desplazamiento y los datos en vivo en lugar de permanecer en una posición de píxel fija. **TsgcChart.RenderContent** pinta los dibujos después de las series y los ejes, recortados al panel de su eje vertical. Junto a los dibujos, una serie lleva sus propios marcadores de precio, **TsgcChartSeries.LastValueMarker** y **TsgcChartSeries.BidAskBand**, ambos `TPersistent` y ambos documentados en esta página.

Propiedades de TsgcChartDrawing

Propiedad	Tipo	Descripción
Kind	TsgcChartDrawingKind	`cdkTrendLine` (el segmento entre los dos anclajes), `cdkHorizontalLine` (una línea a través del panel en Y1), `cdkVerticalLine` (una línea a través del panel en X1), `cdkRectangle` (el cuadro de los dos anclajes, contorno más un relleno ligero) o `cdkFibonacci` (los niveles de retroceso entre X1 y X2). Predeterminado `cdkTrendLine`.
X1, Y1, X2, Y2	Double	Anclajes, como valores de datos de **AxisX** (X) y **AxisY** (Y). Para `cdkFibonacci`, el 0% se sitúa en Y2 y el 100% en Y1 (arrastre desde el inicio del movimiento hasta su final); los niveles abarcan todo el ancho del panel cuando X1 es igual a X2.
AxisX	string	Vacío significa el primer eje inferior, la misma convención que una serie.
AxisY	string	Vacío significa el primer eje izquierdo; el dibujo se pinta en el panel de este eje. Véase Panes.
Color	TsgcUIColor	0 (el valor predeterminado) significa el primer color de la paleta del gráfico.
LineWidth	Double	Píxeles de dispositivo a escala 1. Predeterminado 1.5.
Dash	TsgcChartDrawingDash	`cddSolid`, `cddDash` o `cddDot`. Predeterminado `cddSolid`.
Visible	Boolean	Predeterminado True.
Locked	Boolean	Un dibujo bloqueado todavía se puede seleccionar, pero no mover ni eliminar de forma interactiva. Predeterminado False.

Método de TsgcChartDrawing

Método	Devuelve	Descripción
SetAnchors(const aX1, aY1, aX2, aY2: Double)		Establece los cuatro anclajes con una única notificación de cambio, en lugar de una por propiedad.

TsgcChartDrawings (TsgcChart.Drawings)

TsgcChartDrawings es el `TOwnedCollection` detrás de **TsgcChart.Drawings**.

Miembro	Devuelve	Descripción
Add	TsgcChartDrawing	Añade un dibujo.
AnchorPixels(aIndex: Integer; out aX1, aY1, aX2, aY2: Double; out aPlot: TsgcUIRect)	Boolean	Anclajes del dibujo en aIndex, en píxeles del gráfico, según el último **Render**; también devuelve el rectángulo del panel en el que se pintó el dibujo.
PixelToValue(aIndex: Integer; X, Y: Double; out aX, aY: Double)	Boolean	Valores de datos del píxel de gráfico X, Y en los ejes del dibujo en aIndex, según el último **Render**.
HitTest(X, Y: Double; out aHandle: Integer)	Integer	Índice del dibujo bajo X, Y (píxeles del gráfico), el más superior primero, -1 cuando no hay ninguno. aHandle es el anclaje bajo el punto, 0 o 1, o -1 para un acierto en el cuerpo.
DeleteSelected	Boolean	Elimina **Selected** a menos que esté **Locked**; True cuando se eliminó.
Draw(aCanvas: TsgcUICanvas; aScale: Double)		Pinta cada dibujo visible. Se llama desde **TsgcChart.RenderContent**.
DrawSelection(aCanvas: TsgcUICanvas; aScale: Double)		Pinta los controladores de **Selected**, parte de la superposición de interacción.
SaveToJSON	string	Un objeto con una matriz "drawings", un objeto por dibujo: "kind" (trendline, hline, vline, rectangle o fibonacci), "x1", "y1", "x2", "y2", "axisX", "axisY", "color" (ARGB como un número sin signo), "lineWidth", "dash" (solid, dash o dot), "visible" y "locked". Los números utilizan '.' independientemente de la configuración regional y conservan cada dígito del double.
LoadFromJSON(const aJSON: string)		Sustituye los dibujos (también se acepta una matriz en la raíz); un elemento de un tipo desconocido se omite, y un JSON con formato incorrecto genera **EsgcChartJSONError**.
Selected	Integer (property)	Índice del dibujo seleccionado, -1 cuando no hay ninguno.

Dibujo interactivo y selección

TsgcChartInteraction.DrawingTool crea un dibujo con el ratón: al pulsar se establece el primer anclaje, al arrastrar el segundo, al soltar se termina y la herramienta vuelve a `cdtNone`. Sin ninguna herramienta activa, un clic selecciona el dibujo bajo el puntero (**Drawings.Selected**), un arrastre sobre su cuerpo lo mueve, un arrastre sobre un controlador mueve ese único anclaje, y **TsgcChartInteraction.DeleteSelectedDrawing**, vinculado a la tecla Supr en las vistas VCL y FMX, lo elimina mediante **DeleteSelected**. Véase TsgcChartInteraction para la conexión de ratón y teclado; el código sin vista llama directamente a **Add**, **HitTest** y **Selected**.

TsgcChartLastValueMarker

Una línea discontinua a través del panel en el último valor de una serie, con un cuadro de etiqueta para ese valor sobre el eje vertical, en su lado de trazado.

Propiedad	Tipo	Descripción
Visible	Boolean	Predeterminado False.
Color	TsgcUICo-lor	0 (el valor predeterminado) significa el color de la serie; para una serie de velas, el color de subida o bajada de la última vela.

TsgcChartBidAskBand

Bid y Ask, establecidos por código, dibujados como una banda ligera entre los dos precios, una línea en cada uno y un cuadro de etiqueta para cada uno sobre el eje vertical.

Propiedad	Tipo	Descripción
Visible	Boolean	Predeterminado False.
Bid, Ask	Double	Se establecen juntos con **SetPrices** para generar una única notificación de cambio.
BidColor, AskCo-lor	TsgcUI-Color	Predeterminados a un color de acento verde (bid) y rojo (ask).

SetPrices(const aBid, aAsk: Double) establece ambos precios con una notificación, en lugar de dos.
Ambos marcadores, al igual que los dibujos, se pintan para una serie X/Y (su último Y) y para una serie de velas (su último cierre); una serie muestra uno, ambos o ninguno, independientemente de las demás series del mismo gráfico.

Ejemplo: Una línea de tendencia, el marcador del último valor y una banda bid/ask

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Series_Financial,
  sgcCharts_Drawing, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster;

var
  vChart: TsgcChart;
  vPrice: TsgcChartSeries_Candlestick;
  vTrend: TsgcChartDrawing;
  vAxisBottom, vAxisLeft: TsgcChartAxis;
  vRaster: TsgcUICanvas_Raster;
  i: Integer;
  vClose, vOpen: Double;
  vJSON: string;
  vHandle, vHit: Integer;
  vX1, vY1, vX2, vY2: Double;
  vPlot: TsgcUIRect;
begin
  vChart := TsgcChart.Create(nil);
  try
    { TsgcChart already carries two default axes ('X' and 'Y'); adding named
      ones and keeping a reference to each avoids indexing into Axes, whose
      order includes those defaults }
```

```
    vAxisBottom := vChart.Axes.Add;
    vAxisBottom.Name := 'bottom';
    vAxisBottom.Position := capBottom;

    vAxisLeft := vChart.Axes.Add;
    vAxisLeft.Name := 'left';
    vAxisLeft.Position := capLeft;

    vPrice := TsgcChartSeries_Candlestick.Create(nil);
    try
      vPrice.Chart := vChart;
      vPrice.AxisX := 'bottom';
      vPrice.AxisY := 'left';

      vClose := 100;
      for i := 0 to 79 do
      begin
        vOpen := vClose;
        vClose := vOpen + Sin(i / 6) * 2 + (Random - 0.5);
        vPrice.AddOHLC(i, vOpen, Max(vOpen, vClose) + 1,
          Min(vOpen, vClose) - 1, vClose, 1000 + Random(500));
      end;

      vPrice.LastValueMarker.Visible := True;
      vPrice.BidAskBand.Visible := True;
      vPrice.BidAskBand.SetPrices(vClose - 0.5, vClose + 0.5);

      vTrend := vChart.Drawings.Add;
      vTrend.Kind := cdkTrendLine;
      vTrend.AxisX := 'bottom';
      vTrend.AxisY := 'left';
      vTrend.SetAnchors(10, 95, 70, 105);

      vRaster := TsgcUICanvas_Raster.Create(800, 400);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 800, 400));

        Writeln('drawings: ', vChart.Drawings.Count);
        if vChart.Drawings.AnchorPixels(0, vX1, vY1, vX2, vY2, vPlot) then
        begin
          vHit := vChart.Drawings.HitTest(vX1, vY1, vHandle);
          Writeln('hit test at anchor 1: drawing=', vHit, ' handle=', vHandle);
        end;

        vJSON := vChart.Drawings.SaveToJSON;
        Writeln('JSON length: ', Length(vJSON));
        vChart.Drawings.LoadFromJSON(vJSON);
        Writeln('drawings after reload: ', vChart.Drawings.Count);

        vChart.Drawings.Selected := 0;
        Writeln('deleted: ', vChart.Drawings.DeleteSelected);
        Writeln('drawings after delete: ', vChart.Drawings.Count);

        vRaster.SaveToPNGFile('drawing.png');
      finally
        vRaster.Free;
      end;
    finally
      vPrice.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase También

- TsgcChartSeries_Indicator
- TsgcChartAggregator
- Panes
- TsgcChartInteraction
- TsgcChartSeries_Candlestick
- TsgcChartSeries
- Ediciones

TsgcChartBuffer y TsgcChartOHLCBuffer

Unidad: sgcCharts_Buffer
Búferes circulares para las series del gráfico. **TsgcChartBuffer** guarda puntos X/Y, **TsgcChartOHLCBuffer** guarda X, Open, High, Low, Close y Volume. X es un `Double` (un `TDateTime` se almacena como `Double`). Append es O(1). En modo `cbmOverwrite`, un búfer lleno descarta su punto más antiguo en la siguiente adición; en modo `cbmGrow`, la capacidad se duplica.
Todo método público toma el bloqueo del búfer, así que un hilo de alimentación puede añadir mientras el hilo de render lee. El hilo de render debería llamar a **Snapshot** o **SnapshotRange**, que copian los datos en un registro propiedad del llamador bajo el bloqueo y lo liberan de inmediato, y después decimar y dibujar a partir de esa copia; véase Decimación.
Se espera que X sea monótonamente no decreciente. Append mantiene el indicador **Monotonic**: pasa a False en cuanto llega un punto con una X menor que la anterior, y solo **Clear** lo restablece. Cuando **Monotonic** es True, **SnapshotRange** encuentra el rango visible mediante búsqueda binaria y añade el punto más cercano fuera de cada borde, de modo que se dibujan las líneas que salen del trazado. Cuando es False, **SnapshotRange** recurre a un recorrido lineal que solo copia los puntos dentro del rango, y las funciones de decimación devuelven los datos sin decimar.

Propiedades y métodos compartidos (TsgcChartBufferBase)

Miembro	Descripción
Capacity: Integer	Capacidad actual en puntos o velas. Aumentarla reasigna memoria; reducirla por debajo de **Count** descarta los datos más antiguos.
Count: Integer	Número de puntos o velas contenidos actualmente.
Mode: TsgcChartBufferMode	`cbmOverwrite` (descartar el punto más antiguo cuando está lleno) o `cbmGrow` (duplicar la capacidad cuando está lleno).
Monotonic: Boolean	True mientras cada X añadida haya sido >= la X anterior. Véase más arriba.
Lock / Unlock	Toma o libera el propio bloqueo del búfer, para un llamador que necesita que más de una llamada sea atómica.
Clear	Vacía el búfer y restablece **Monotonic** a True.

TsgcChartBuffer

Miembro	Descripción
Create(aCapacity: Integer = 1024; aMode: TsgcChartBufferMode = cbmOverwrite)	Constructor.
Append(const aX, aY: Double)	Añade un punto.
AppendBatch(const aX, aY: array of Double)	Añade varios puntos en una sola llamada bloqueada.
First(out aX, aY: Double): Boolean	Punto más antiguo. False cuando el búfer está vacío.

Last(out aX, aY: Double): Boolean	Punto más reciente. False cuando el búfer está vacío.
Snapshot(var aSnap: TsgcChartBufferSnapshot)	Copia cada punto en `aSnap`. Los arrays se reutilizan entre llamadas y pueden ser más largos que `aSnap.Count`.
SnapshotRange(const aXMin, aXMax: Double; var aSnap: TsgcChartBufferSnapshot)	Copia los puntos visibles en [aXMin, aXMax], más los puntos de borde descritos arriba cuando **Monotonic**.
X[aIndex], Y[aIndex]: Double	Orden lógico: 0 es el punto más antiguo, `Count - 1` el más reciente.

TsgcChartOHLCBuffer

Miembro	Descripción
Create(aCapacity: Integer = 1024; aMode: TsgcChartBufferMode = cbmOverwrite)	Constructor.
Append(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)	Añade una vela.
AppendBatch(const aX, aOpen, aHigh, aLow, aClose, aVolume: array of Double)	Añade varias velas en una sola llamada bloqueada. `aVolume` puede estar vacío, el volumen es entonces 0.
First / Last(out aX, aOpen, aHigh, aLow, aClose, aVolume: Double): Boolean	Vela más antigua / más reciente. False cuando el búfer está vacío.
UpdateLast(const aX, aOpen, aHigh, aLow, aClose, aVolume: Double): Boolean	Sustituye los valores de la vela más reciente cuando su X es igual a `aX`, para una vela en vivo aún en formación. Devuelve False y no cambia nada cuando el búfer está vacío o la vela más reciente tiene otra X.
Snapshot(var aSnap: TsgcChartOHLCSnapshot)	Copia cada vela en `aSnap`.
SnapshotRange(const aXMin, aXMax: Double; var aSnap: TsgcChartOHLCSnapshot)	Copia las velas visibles en [aXMin, aXMax].
X[aIndex], Open[aIndex], High[aIndex], Low[aIndex], Close[aIndex], Volume[aIndex]: Double	Orden lógico: 0 es la vela más antigua, `Count - 1` la más reciente.

Ejemplo: añadir datos y tomar una instantánea

```
var
  vBuffer: TsgcChartBuffer;
  vSnap: TsgcChartBufferSnapshot;
  i: Integer;
begin
  vBuffer := TsgcChartBuffer.Create(1024, cbmOverwrite);
  try
    for i := 0 to 999 do
      vBuffer.Append(i, Sin(i / 20) * 50);

    vBuffer.Snapshot(vSnap);
    Writeln(Format('points: %d  first x: %g  last x: %g',
```

```
      [vSnap.Count, vSnap.X[0], vSnap.X[vSnap.Count - 1]]));
  finally
    vBuffer.Free;
  end;
end;
```

Véase también

- Decimación
- TsgcChartDataSource

Decimación

Unidad: sgcCharts_Buffer
Funciones sin dependencia de ningún framework que reducen una instantánea de búfer a aproximadamente un conjunto de puntos por columna de píxeles, para un trazado de `aPixelWidth` píxeles de ancho que muestra `[aXMin, aXMax]`. Una instantánea no monótona, un rango vacío, o `aPixelWidth < 1` se copia sin decimar en todos los casos siguientes.

Valor / Función	Descripción
TsgcChartDecimation = (dmNone, dmMinMax, dmLTTB)	`dmNone` copia cada punto. `dmMinMax` divide X en columnas de píxeles y conserva, para cada columna, el primer, mínimo, máximo y último punto en orden de X (como máximo 4), de modo que la polilínea se mantiene visualmente exacta; los puntos a la izquierda de `aXMin` comparten una columna adicional, los puntos a la derecha de `aXMax` otra. `dmLTTB` devuelve `2 × aPixelWidth` puntos elegidos mediante Largest Triangle Three Buckets, pensado para series estáticas.
sgcChartDecimate(const aSnap: TsgcChartBufferSnapshot; const aXMin, aXMax: Double; aPixelWidth: Integer; aMode: TsgcChartDecimation; var aOutX, aOutY: TsgcChartDoubleArray; out aOutCount: Integer)	Decima una instantánea de puntos usando `aMode`. `aOutX` y `aOutY` crecen cuando son demasiado cortos, nunca se reducen.
sgcChartLTTB(const aX, aY: TsgcChartDoubleArray; aCount, aThreshold: Integer; var aOutX, aOutY: TsgcChartDoubleArray; out aOutCount: Integer)	Largest Triangle Three Buckets directamente sobre un par de arrays X/Y. Devuelve exactamente `Min(aThreshold, aCount)` puntos, incluyendo siempre el primero y el último.
sgcChartDecimateOHLC(const aSnap: TsgcChartOHLCSnapshot; const aXMin, aXMax: Double; aPixelWidth: Integer; var aOut: TsgcChartOHLCSnapshot)	Fusiona las velas que caen en la misma columna de píxeles: X y apertura de la primera, máximo más alto, mínimo más bajo, cierre de la última, suma de los volúmenes. `aOut.Count` contiene el resultado.

Ejemplo: decimar una instantánea para un trazado de 800px

```
var
  vBuffer: TsgcChartBuffer;
  vSnap: TsgcChartBufferSnapshot;
  vOutX, vOutY: TsgcChartDoubleArray;
  vOutCount, i: Integer;
begin
  vBuffer := TsgcChartBuffer.Create(100000, cbmOverwrite);
  try
    for i := 0 to 99999 do
      vBuffer.Append(i, Sin(i / 200) * 100);

    vBuffer.Snapshot(vSnap);
    sgcChartDecimate(vSnap, 0, 99999, 800, dmMinMax, vOutX, vOutY, vOutCount);
    Writeln(Format('decimated from %d to %d points', [vSnap.Count, vOutCount]));
  finally
    vBuffer.Free;
  end;
end;
```

Véase también

- TsgcChartBuffer y TsgcChartOHLCBuffer

TsgcUICanvas

Unidad: sgcUICore_Canvas, con dos backends: sgcUICore_Canvas_Raster (**TsgcUICanvas_Raster**) y sgcUICore_Canvas_SVG (**TsgcUICanvas_SVG**)
TsgcUICanvas es la superficie de dibujo abstracta sobre la que pinta TsgcChart.**Render**. El paquete básico incluye dos backends concretos: **TsgcUICanvas_Raster** pinta en un búfer ARGB en memoria y lo guarda como PNG, **TsgcUICanvas_SVG** construye un documento SVG. Ambos comparten la misma API de trazado, pintura y texto, así que un gráfico se renderiza de forma idéntica con cualquiera de los dos. Una aplicación normalmente solo crea el lienzo, llama a **Render**, y guarda el resultado; las primitivas de dibujo de abajo también son públicas para añadir superposiciones personalizadas.

Tipos

Tipo	Descripción
TsgcUIColor = Cardinal	`$AARRGGBB` directo (no premultiplicado).
TsgcUIRect	`Left`, `Top`, `Right`, `Bottom`: Double, píxeles de dispositivo; Right / Bottom son los bordes lejanos.
TsgcUIFontStyle	`Size`: Double, `Bold`: Boolean, `Color`: TsgcUIColor.
TsgcUILineJoin, TsgcUILineCap	`(ljMiter, ljRound, ljBevel)`, `(lcButt, lcRound, lcSquare)`.
TsgcUITextHAlign, TsgcUITextVAlign	`(thaLeft, thaCenter, thaRight)`, `(tvaTop, tvaMiddle, tvaBaseline, tvaBottom)`.
TsgcUIFillRule	`frNonZero`, `frEvenOdd`.

Métodos

Método	Descripción
Clear(aColor: TsgcUIColor)	Rellena toda la superficie.
BeginPath, MoveTo, LineTo, QuadTo, CubicTo, Arc, RoundRect, Rectangle, ClosePath	Construcción de trazados. **Arc** recibe ángulos en radianes, 0 en +X, barrido positivo en sentido horario en pantalla.
FillPath(aColor: TsgcUIColor; aRule: TsgcUIFillRule = frNonZero)	Rellena el trazado actual; se conserva hasta el siguiente **BeginPath**.
FillPathLinearGradient(aX1, aY1, aX2, aY2: Double; const aStops: array of TsgcUIGradientStop; aRule: TsgcUIFillRule = frNonZero)	Rellena el trazado actual con un degradado lineal.
StrokePath(aColor, aWidth[, aJoin, aCap[, aDash, aDashOffset]])	Dos sobrecargas: con o sin patrón de trazos.
DrawText, MeasureText, TextWidth, LineHeight, FitText	Dibujo y medición de texto con un **TsgcUIFontStyle**. Véase Fuentes.
DrawImage(const aImage: TsgcUIImage; aX, aY: Integer; aOpacity: Double = 1)	Vuelca una imagen ARGB.

PushClip(aLeft, aTop, aRight, aBottom: Integer), PopClip	Rectángulo de recorte en píxeles de dispositivo, Right / Bottom exclusivos; los recortes anidados se intersecan.
SetTransform(aA, aB, aC, aD, aE, aF: Double), ResetTransform	Transformación afín: `x' = A*x + C*y + E`, `y' = B*x + D*y + F`.
SetFontFile(const aRegular, aBold: string), GetFont(aBold: Boolean)	Fuentes usadas para medir y dibujar texto; un nombre de archivo vacío selecciona la Roboto incrustada. Véase Fuentes.
Width, Height	Tamaño de la superficie en píxeles de dispositivo.

Adiciones de TsgcUICanvas_Raster

Miembro	Descripción
Create(aWidth, aHeight: Integer)	Sobrecarga que reserva su propio rasterizador.
Create(aRasterizer: TsgcUIRasterizer)	Sobrecarga que dibuja sobre un rasterizador existente.
SetSize(aWidth, aHeight: Integer)	Redimensiona la superficie.
SaveToPNGStream(aStream: TStream; aCompress: Boolean = True)	Codifica la superficie como PNG.
SaveToPNGFile(const aFileName: string; aCompress: Boolean = True)	Codifica y escribe el PNG en disco.
Rasterizer	TsgcUIRasterizer (solo lectura), el rasterizador subyacente.

Adiciones de TsgcUICanvas_SVG

Miembro	Descripción
Create(aWidth, aHeight: Integer)	Crea un documento vacío del tamaño indicado.
SetSize(aWidth, aHeight: Integer)	Nuevo documento vacío del tamaño indicado.
SaveToString	string, el documento completo con la declaración XML; los recortes abiertos se cierran en la salida.
SaveToStream(aStream: TStream), SaveToFile(const aFileName: string)	Escribe el documento.

Renderizar un gráfico a PNG o SVG

Cree el lienzo con el tamaño de destino, llame a **TsgcChart.Render** con un **TsgcUIRect** que lo cubra, y después guarde. Esta es toda la ruta de código que necesita un informe sin interfaz visual o una exportación programada, sin ningún control VCL ni FireMonkey involucrado.

Ejemplo: el mismo gráfico renderizado a PNG y a SVG

```
uses
  SysUtils, Math,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_Axis, sgcCharts_Types,
  sgcUICore_Canvas, sgcUICore_Canvas_Raster, sgcUICore_Canvas_SVG;

var
  vChart: TsgcChart;
  vSeries: TsgcChartSeries_Line;
  i: Integer;
  vRaster: TsgcUICanvas_Raster;
  vSVG: TsgcUICanvas_SVG;
begin
  vChart := TsgcChart.Create(nil);
  try
    vChart.Axes.Add.Position := capBottom;
    vChart.Axes.Add.Position := capLeft;

    vSeries := TsgcChartSeries_Line.Create(nil);
    try
      vSeries.Chart := vChart;
      for i := 0 to 59 do
        vSeries.AddXY(i, Sin(i / 8) * 30 + 50);

      vRaster := TsgcUICanvas_Raster.Create(500, 250);
      try
        vChart.Render(vRaster, sgcUIRect(0, 0, 500, 250));
        vRaster.SaveToPNGFile('canvas.png');
      finally
        vRaster.Free;
      end;

      vSVG := TsgcUICanvas_SVG.Create(500, 250);
      try
        vChart.Render(vSVG, sgcUIRect(0, 0, 500, 250));
        Writeln('SVG length: ', Length(vSVG.SaveToString));
        vSVG.SaveToFile('canvas.svg');
      finally
        vSVG.Free;
      end;
    finally
      vSeries.Free;
    end;
  finally
    vChart.Free;
  end;
end;
```

Véase también

- TsgcChart
- Fuentes

TsgcUIFont

Unidad: sgcUICore_Font
TsgcUIFont es el motor de texto TrueType en Pascal puro que usa todo TsgcUICanvas para medir y dibujar texto; analiza el archivo de fuente por sí mismo, sin depender de la pila de texto del sistema operativo. Cada lienzo crea su propio **TsgcUIFont** normal y en negrita a través de **SetFontFile** / **GetFont**, así que una aplicación rara vez crea uno directamente salvo para medir texto fuera de un lienzo.

Propiedades

Propiedad	Tipo	Descripción
Loaded	Boolean (solo lectura)	True una vez se ha analizado un archivo o flujo de fuente.
UnitsPerEm, Ascender, Descender, LineGap	Integer (solo lectura)	Métricas de la fuente en unidades de fuente.
GlyphCount	Integer (solo lectura)	
CmapFormat	Integer (solo lectura)	El formato de la tabla de carácter a glifo que se encuentra en el archivo.
HasKerning	Boolean (solo lectura)	True solo cuando el archivo de fuente cargado tiene una tabla kern propia.
Kerning	Boolean	Valor predeterminado True. Aplica la tabla kern de la fuente, cuando la tiene, durante la medición y el dibujo; establézcalo en False para desactivar el interletraje incluso en una fuente que tenga tabla.
CacheCapacity, CacheCount	Integer	Tamaño y número actual de entradas de la caché interna de contornos de glifo.

Métodos

Método	Devuelve	Descripción
LoadFromFile(const aFileName: string)		Analiza un archivo de fuente TrueType (contorno glyf). Genera **EsgcUIFont** para una fuente CFF (contorno PostScript), o cualquier otra cosa que no sea una fuente TrueType utilizable.
LoadFromStream(aStream: TStream)		Lo mismo, desde un flujo.
LoadDefault(aBold: Boolean = False)		Carga la fuente incrustada en sgcCharts (Roboto Regular o Bold). Siempre tiene éxito; no necesita ningún archivo.
LoadPlatformDefault(aBold: Boolean = False)		Mejor intento de fuente de interfaz del sistema para la plataforma actual (Segoe UI en Windows, Roboto en Android, DejaVu Sans o Liberation Sans en Linux, Helvetica o Arial en macOS/iOS); recurre a **LoadDefault** cuando no

		se encuentra ninguno de esos archivos, así que nunca genera una excepción.
CharToGlyph(aCode: Cardinal)	Integer	Punto de código Unicode a id de glifo; 0 (.notdef) cuando no está asignado.
GlyphAdvance(aGlyph: Integer)	Integer	Ancho de avance en unidades de fuente.
GetKerning(aLeft, aRight: Integer)	Integer	Ajuste de la tabla kern en unidades de fuente para un par de glifos; 0 sin un par.
IsCompositeGlyph(aGlyph: Integer)	Boolean	
GetGlyphOutline(aGlyph: Integer)	TsgcUIGlyphOutline	Contorno en caché en unidades de fuente, propiedad de la fuente, válido hasta la siguiente llamada que pueda desalojarlo; nil para un id de glifo no válido.
MeasureText(const aText: string; aSize: Double; out aWidth, aAscent, aDescent: Double)		
TextWidth(const aText: string; aSize: Double)	Double	
LineHeight(aSize: Double)	Double	
FitText(const aText: string; aSize, aMaxWidth: Double)	string	Trunca **aText** para que quepa en **aMaxWidth**.
DrawText(aRaster: TsgcUIRasterizer; const aText: string; aX, aY: Double; aSize: Double; aColor: Cardinal; aAngle: Double = 0)		Dibuja directamente sobre un rasterizador, por debajo del nivel de **TsgcUICanvas**.
TextToCodePoints(const aText: string) (class function)	TsgcUICodePointArray	

Fuente incrustada y fuentes de la plataforma

sgcCharts incrusta una compilación TrueType de Roboto para que todo gráfico dibuje texto sin ningún archivo externo: **LoadDefault** la carga directamente, y es lo que selecciona un nombre de archivo vacío pasado a **TsgcUICanvas.SetFontFile**. **LoadPlatformDefault** prueba primero la propia fuente de interfaz de la plataforma y solo recurre a la incrustada cuando no puede encontrar o analizar un archivo utilizable, así que es la opción predeterminada más segura para una aplicación que quiera coincidir con el sistema operativo anfitrión. La compilación de Roboto incrustada no lleva tabla kern, así que **HasKerning** devuelve False para ella aunque **Kerning** siga en True; una fuente TrueType personalizada cargada con su propia tabla kern se interletra automáticamente.

Solo se admiten contornos TrueType

TsgcUIFont lee directamente la tabla de contorno `glyf`; no tiene analizador CFF (contorno PostScript). Muchas fuentes del sistema que parecen archivos TrueType simples, como Helvetica o SF en macOS, son en realidad CFF, y **LoadFromFile** / **LoadFromStream** generan **EsgcUIFont** para ellas. Cuando una fuente predeterminada de la plataforma resulta ser CFF, **LoadPlatformDefault** lo detecta y pasa al siguiente candidato, terminando en la fuente incrustada si no se encuentra nada utilizable.

Ejemplo: fuente incrustada, fuente de la plataforma y medición de texto

```
uses
  SysUtils,
  sgcUICore_Font;

var
  vFont: TsgcUIFont;
  vWidth, vAscent, vDescent: Double;
begin
  vFont := TsgcUIFont.Create;
  try
    vFont.LoadDefault(False);
    Writeln('embedded Roboto loaded: ', vFont.Loaded);
    Writeln('embedded Roboto has kerning: ', vFont.HasKerning);

    vFont.MeasureText('sgcCharts', 14, vWidth, vAscent, vDescent);
    Writeln(Format('width=%.2f ascent=%.2f descent=%.2f',
      [vWidth, vAscent, vDescent]));
  finally
    vFont.Free;
  end;

  vFont := TsgcUIFont.Create;
  try
    vFont.LoadPlatformDefault(False);
    Writeln('platform font loaded: ', vFont.Loaded);
    Writeln('platform font glyph count: ', vFont.GlyphCount);
  finally
    vFont.Free;
  end;
end;
```

Véase también

- Lienzo

Indicadores

Unidad: sgcCharts_Indicators *(solo edición financiera, compilada bajo `SGC_CHARTS_FINANCIAL`; una envoltura vacía en caso contrario)*
Calculadores de indicadores técnicos incrementales, sin dependencia de ningún framework: **TsgcChartIndicatorSMA**, **EMA**, **WMA**, **Bollinger**, **RSI**, **MACD**, **Stochastic**, **ATR** y **VWAP**, todos descendientes de **TsgcChartIndicatorCalc**. No dependen de TsgcChart y pueden alimentar una serie adicional, un motor de señales, o código de informes simple.

Patrón común

Miembro	Descripción
Add(...)	Alimenta una barra nueva y devuelve la última salida. O(1) amortizado, excepto Stochastic, que recorre su ventana de máximos/mínimos (O(KPeriod)).
UpdateLast(...)	Sustituye la entrada de la barra añadida más recientemente, para una vela en vivo aún en formación: se restaura el estado guardado antes del último **Add** y se añade de nuevo el nuevo valor, de modo que el resultado coincide con un cálculo desde cero con la barra modificada. En un calculador vacío se comporta como **Add**.
Calculate(const aInput: array of Double; ...)	Reinicia el calculador y alimenta un array completo a través de **Add**, de modo que los resultados por lotes e incrementales coinciden. El calculador conserva su estado después, así que se pueden añadir barras en vivo tras cargar un histórico.
Reset	Borra todo el estado; **Count** vuelve a 0.
Count	Integer (solo lectura). Barras añadidas desde el último **Reset**.

Periodo de calentamiento

Toda salida es `NaN` (`Math.NaN`, compruébelo con `Math.IsNan`) hasta que termina el periodo de calentamiento del calculador; una entrada `NaN` se ignora, así que no entra en la ventana y la salida de esa barra también es `NaN`. Los índices de barra de abajo son de base 0.

Definiciones

Calculador	Constructor	Definición
TsgcChartIndicatorSMA	Create(aPeriod: Integer = 20)	Media aritmética de los últimos **Period** valores; primer valor en la barra Period-1.
TsgcChartIndicatorEMA	Create(aPeriod: Integer = 20)	alpha = 2/(Period+1), sembrado con la SMA de los primeros Period valores en la barra Period-1, después Value := X * alpha + Prev * (1 - alpha).
TsgcChartIndicatorWMA	Create(aPeriod: Integer = 20)	Media móvil ponderada linealmente, pesos 1..Period (Period en la más reciente), dividida entre Period(Period+1)/2; primer valor en la barra Period-1.

TsgcChartIndicatorBollinger	Create(aPeriod: Integer = 20; const aDeviations: Double = 2)	Middle = SMA(Period), desviación estándar poblacional (dividir entre Period), Upper / Lower = Middle +/- Deviations * StdDev; primer valor en la barra Period-1.
TsgcChartIndicatorRSI	Create(aPeriod: Integer = 14)	Suavizado de Wilder. Change(i) = Close(i) - Close(i-1); la primera ganancia/pérdida media es la media de los primeros Period cambios (barra Period), después Avg := (Avg * (Period-1) + Current) / Period. RSI = 100 - 100 / (1 + AvgGain / AvgLoss); 100 cuando AvgLoss es 0, 50 cuando ambos son 0.
TsgcChartIndicatorMACD	Create(aFastPeriod: Integer = 12; aSlowPeriod: Integer = 26; aSignalPeriod: Integer = 9)	Line = EMA(Fast) - EMA(Slow), primera Line en la barra Slow-1. Signal = EMA(Signal) de la Line, sembrada con la SMA de los primeros valores de Line de Signal, primera en la barra Slow+Signal-2. Histogram = Line - Signal.
TsgcChartIndicatorStochastic	Create(aKPeriod: Integer = 14; aKSmoothing: Integer = 3; aDPeriod: Integer = 3)	Raw = 100 * (Close - LowestLow) / (HighestHigh - LowestLow) sobre los últimos máximos/mínimos de KPeriod (50 cuando HighestHigh = LowestLow), primero en la barra KPeriod-1. K = SMA(KSmoothing) de Raw (KSmoothing = 1 es el estocástico rápido), D = SMA(DPeriod) de K.
TsgcChartIndicatorATR	Create(aPeriod: Integer = 14)	Wilder. TR(i) = Max(High-Low, |High-PrevClose|, |Low-PrevClose|) desde la barra 1. El primer ATR es la media de los primeros Period TR (barra Period), después ATR := (ATR * (Period-1) + TR) / Period.
TsgcChartIndicatorVWAP	Create(const aSessionStart: TDateTime = 0)	Sum(TypicalPrice * Volume) / Sum(Volume) acumulado, TypicalPrice = (High+Low+Close)/3. Las sumas se reinician en la primera barra de cada sesión; la sesión de una barra es Floor(Time - SessionStart), así que SessionStart = 0 reinicia cada día natural y 9.5/24 reinicia a las 09:30. Sin periodo de calentamiento; NaN mientras el volumen de la sesión es 0.

Ejemplo: SMA y RSI, en flujo y por lotes

```
{$IFDEF SGC_CHARTS_FINANCIAL}
uses
  SysUtils, Math,
  sgcCharts_Buffer, sgcCharts_Indicators;

var
  vSMA: TsgcChartIndicatorSMA;
  vRSI: TsgcChartIndicatorRSI;
  vCloses: TsgcChartDoubleArray;
  vOut: TsgcChartDoubleArray;
  i: Integer;
  vValue: Double;
begin
  SetLength(vCloses, 40);
  for i := 0 to 39 do
    vCloses[i] := 100 + Sin(i / 4) * 5;

  vSMA := TsgcChartIndicatorSMA.Create(10);
  try
    for i := 0 to High(vCloses) do
    begin
      vValue := vSMA.Add(vCloses[i]);
      if (i = High(vCloses)) and not IsNan(vValue) then
        Writeln(Format('SMA(10) last = %.4f (count=%d)', [vValue, vSMA.Count]));
    end;

    vSMA.Calculate(vCloses, vOut);
    Writeln('batch SMA matches streaming last: ',
```

```
      SameValue(vOut[High(vOut)], vSMA.Value, 1E-9));
  finally
    vSMA.Free;
  end;

  vRSI := TsgcChartIndicatorRSI.Create(14);
  try
    for i := 0 to High(vCloses) do
      vValue := vRSI.Add(vCloses[i]);
    Writeln(Format('RSI(14) last = %.2f', [vRSI.Value]));
  finally
    vRSI.Free;
  end;
end;
{$ENDIF}
```

Véase también

- TsgcChartSeries

Información general de .NET

sgcCharts .NET es un puerto en C# gestionado de la biblioteca Delphi sgcCharts, no un envoltorio sobre ella: cada unidad Delphi tiene su contrapartida en C# bajo el espacio de nombres **esegece.sgcCharts**, usando los mismos nombres de clase **Tsgc***, de modo que el código escrito contra la API de Delphi se lee igual en C#. El puerto se divide en seis paquetes NuGet, así que un proyecto solo incorpora la plataforma a la que se dirige.

Paquetes

PackageId	Contiene	Frameworks de destino
esegece.sgcCharts	El modelo de gráfico básico: **TsgcChart**, ejes, paneles, series, fuentes de datos, búferes, dibujos, las funciones matemáticas y de formato, y los ayudantes JSON, y el lienzo SVG. Sin dependencia nativa.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcCharts.Rendering	**TsgcUICanvas_Raster**, el lienzo rasterizado de SkiaSharp usado para la salida PNG sin interfaz visual en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcCharts.WinForms	**TsgcWinFormsChart**, la vista de Windows Forms.	net472, net8.0-windows, net9.0-windows
esegece.sgcCharts.Wpf	**TsgcWpfChart**, la vista WPF.	net472, net8.0-windows, net9.0-windows
esegece.sgcCharts.Avalonia	**TsgcAvaloniaChart**, la vista multiplataforma (Windows, Linux, macOS).	net8.0, net9.0
esegece.sgcCharts.sgcWebSockets	El puente en vivo: fuentes de datos de gráfico alimentadas desde una conexión **sgcWebSockets** (mensajes WebSocket, publicaciones MQTT, server-sent events, flujos de mercado de Binance). Depende del paquete NuGet **esegece.sgcWebSockets**.	netstandard2.0, net8.0, net9.0

Ediciones Community y Registered

La compilación predeterminada de cada paquete anterior (sin ningún interruptor) es la edición de pago **Registered**: sin constante `SGC_COMMUNITY`, sin logotipo incrustado, sin aviso de inicio. Compilar o empaquetar con `-p:SgcCommunity=true` produce en su lugar la edición **Community** (gratuita), añadiendo un sufijo **.Community** al id de paquete de todos los paquetes excepto el puente (**esegece.sgcCharts.sgcWebSockets**), que la edición Community no incluye en absoluto, la misma regla que sigue la edición Community de Delphi para sus propias unidades de puente. La compilación Community conserva las ocho series, con velas y OHLC incluidas, y todas las vis-

tas, y solo excluye el paquete financiero (paneles, series de indicadores y volumen, agregador, herramientas de dibujo, marcador de último valor y banda de oferta/demanda).
Community es gratuita tanto para uso comercial como no comercial de equipos pequeños: personas y organizaciones con ingresos brutos anuales inferiores a 1.000.000 EUR y no más de 5 desarrolladores. Las organizaciones que igualen o superen cualquiera de los dos umbrales necesitan una licencia Comercial (Registered).

La API refleja Delphi

Las clases de vista conservan la nomenclatura de Delphi: **TsgcWinFormsChart** y **TsgcWpfChart** son las contrapartidas en .NET del **TsgcVCLChart** de Delphi, y **TsgcAvaloniaChart** es la contrapartida del **TsgcFMXChart** de Delphi.
Las notificaciones del modelo básico, como **TsgcChart.OnChange**, **TsgcChartDataSource.OnDataChanged**, **TsgcChartDataSource.OnBeforeAppend**, **TsgcChart.OnCrosshairMove**, **OnSeriesClick** y **OnZoom**, son simples propiedades delegadas `get; set;`, no miembros `event` de C#. Asignar un nuevo manejador sustituye al anterior, exactamente igual que un puntero a método simple de Delphi. El único lugar donde interviene un auténtico evento multidifusión de C# (`+=` / `-=`) es donde el puente **esegece.sgcCharts.sgcWebSockets** se suscribe a un cliente del puerto .NET de sgcWebSockets; véase Bridge Overview y Diferencias de .NET respecto a Delphi.
La mayoría de las propiedades de array indexadas de Delphi se convierten en auténticos indexadores de C#: **TsgcChartAxes**, **TsgcChartPanes**, **TsgcChartDrawings** y **TsgcChartPalette** exponen todas `this[int Index]`, y **TsgcChartJSONNode** expone `this[int aIndex]` para sus elementos de array. Los búferes de puntos que hay detrás de una serie son la excepción: **TsgcChartBuffer.X(Index)** / **Y(Index)**, **TsgcChartOHLCBuffer.X(Index)** / **Open(Index)** / **High(Index)** / **Low(Index)** / **Close(Index)** / **Volume(Index)**, y **XValues(Index)** / **Values(Line, Index)** de las series financieras son métodos normales, porque C# solo permite una firma de indexador por clase y cada una de estas necesita varios accesores por índice, con nombre propio o con varios parámetros, en la misma clase. Véase Diferencias de .NET respecto a Delphi.

Vistas y lienzos desechables

Las clases que mantienen una superficie Skia nativa persistente o un mapa de bits GDI dimensionado según el control implementan `IDisposable` y deben liberarse, o su memoria nativa tendrá fugas.

- **TsgcUICanvas_Raster** (esegece.sgcCharts.Rendering): posee el mapa de bits, el lienzo, los pinceles y las tipografías de SkiaSharp.
- **TsgcChartRenderer**: posee su lienzo rasterizado.
- **TsgcWinFormsChart**: libera su lienzo rasterizado, su mapa de bits Skia y su mapa de bits GDI+, y su temporizador de limitación, desde un `Dispose(bool)` sobrescrito.
- **TsgcWpfChart**: libera su lienzo rasterizado y su mapa de bits Skia.
- El fotograma interno de **TsgcAvaloniaChart** (**TsgcAvaloniaChartFrame**): libera sus imágenes Skia frontal y trasera y sus lienzos rasterizados.

El propio modelo de gráfico básico (**TsgcChart**, series, fuentes de datos) no mantiene ningún recurso nativo y no necesita liberarse por ese motivo; aun así implementa `IDisposable` en algunos lugares para ejecutar una notificación de liberación al estilo Delphi, como desvincular una serie de su fuente o una fuente de datos de su cliente.

Métrica de texto idéntica a Delphi

Todo lienzo mide el texto con **TsgcUIFont**, un lector TrueType portado de la parte METRICS del `sgcUICore_Font.pas` de Delphi: lee una tabla `cmap` de formato 4 o 12 para la búsqueda de carácter a glifo, los anchos de avance de `hmtx`, los pares de interletraje de una subtabla `kern` de formato 0, y el ascendente, el descendente y el espacio entre líneas de `hhea`. Carga los mismos archivos incrustados **Roboto-Regular.ttf** y **Roboto-Bold.ttf** que incrusta la biblioteca Delphi (Apache License 2.0), así que un diseño calculado en .NET coincide con el diseño de Delphi hasta el último bit. SkiaSharp se usa solo para rasterizar los contornos de los glifos que el lector de métricas ya ha posicionado, un glifo posicionado por carácter, con suavizado de bordes y posicionamiento subpíxel sin hinting; nunca mide el texto por sí mismo, que es lo que mantiene a todo backend, tanto rasterizado como SVG, alineado con Delphi.

Véase también

- Diferencias de .NET respecto a Delphi
- TsgcChart
- TsgcVCLChart
- Bridge Overview

Inicio Rápido: WinForms

TsgcWinFormsChart es la vista de Windows Forms de un modelo TsgcChart: un descendiente de `Control` que pinta el gráfico asignado a su propiedad **Chart**. Se incluye en el paquete **esegece.sgcCharts.WinForms**, destinado a `net472`, `net8.0-windows` y `net9.0-windows`, y lleva de forma transitiva una `ProjectReference` (o una `PackageReference`, cuando se consume desde NuGet) a **esegece.sgcCharts.Rendering**, así que un proyecto WinForms solo necesita referenciar el propio **esegece.sgcCharts.WinForms**. El control se dibuja con el mismo lienzo rasterizado de Skia que el TsgcVCLChart de Delphi, directamente en una sección DIB de GDI persistente, limitado a **Chart.MaxFPS**, y es seguro alimentarlo con datos desde un hilo en segundo plano.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este control. Asignarlo registra el control como oyente; borrarlo, o liberar el gráfico, hace que el control deje de pintar nada.
FrameCount	uint (solo lectura)	Número de fotogramas completos renderizados desde que se creó el control.
OverlayFrameCount	uint (solo lectura)	Número de fotogramas solo de superposición renderizados desde que se creó el control, para un cambio que solo mueve la retícula, la información sobre herramientas o una selección de banda elástica.

Las propiedades habituales de `Control` también son públicas y no cambian: **Dock**, **Anchor**, **BackColor**, **Enabled**, **Visible**, **TabIndex**, **TabStop**, **ContextMenuStrip**, y los eventos de ratón, arrastre y redimensionado.

Métodos

Método	Devuelve	Descripción
SaveToPNG(string aFileName)	void	Renderiza **Chart** al tamaño de cliente y DPI actuales del control en un nuevo archivo PNG. No hace nada cuando **Chart** es null.
SaveToSVG(string aFileName)	void	Renderiza **Chart** al tamaño de cliente y DPI actuales del control en un archivo SVG. No hace nada cuando **Chart** es null.

Ejemplo: representar un gráfico en un formulario WinForms

Véase también

- Información general de .NET
- Inicio Rápido: WPF
- TsgcVCLChart
- TsgcChartSeries_Line

Inicio Rápido: WPF

TsgcWpfChart es la vista WPF de un modelo TsgcChart: un descendiente de `Control` que pinta el gráfico asignado a su propiedad de dependencia **Chart**. Se incluye en el paquete **esegece.sgcCharts.Wpf**, destinado a `net472`, `net8.0-windows` y `net9.0-windows`, y lleva de forma transitiva una `ProjectReference` (o una `PackageReference`, cuando se consume desde NuGet) a **esegece.sgcCharts.Rendering**, así que un proyecto WPF solo necesita referenciar el propio **esegece.sgcCharts.Wpf**. El control se dibuja con el mismo lienzo rasterizado de Skia que el TsgcVCLChart de Delphi, directamente en el back buffer de un `WriteableBitmap` persistente, limitado a **Chart.MaxFPS**, y es seguro alimentarlo con datos desde un hilo en segundo plano.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este control, respaldado por la propiedad de dependencia **ChartProperty**. Asignarlo registra el control como oyente mientras está cargado; borrarlo, o liberar el gráfico, hace que el control deje de pintar nada.
FrameCount	long (solo lectura)	Número de fotogramas completos renderizados desde que se creó el control.
OverlayFrameCount	long (solo lectura)	Número de fotogramas solo de superposición renderizados desde que se creó el control, para un cambio que solo mueve la retícula, la información sobre herramientas o una selección de banda elástica.

Las propiedades habituales de `Control` también son públicas y no cambian: **Width**, **Height**, **HorizontalAlignment**, **VerticalAlignment**, **Background**, **IsEnabled**, **Visibility**, **Focusable**, y los eventos enrutados de ratón y teclado.

Métodos

Método	Devuelve	Descripción
SaveToPNG(string aFileName)	void	Renderiza **Chart** al tamaño y DPI de render actuales del control en un nuevo archivo PNG. No hace nada cuando **Chart** es null.
SaveToSVG(string aFileName)	void	Renderiza **Chart** al tamaño y DPI de render actuales del control en un archivo SVG. No hace nada cuando **Chart** es null.

Ejemplo: representar un gráfico en una ventana WPF

Véase también

- Información general de .NET
- Inicio Rápido: WinForms
- TsgcVCLChart
- TsgcChartSeries_Line

Inicio Rápido: Avalonia

Paquete: esegece.sgcCharts.Avalonia (destinado a net8.0 y net9.0, depende de **esegece.sgcCharts.Rendering**)
TsgcAvaloniaChart es la vista Avalonia de un modelo **TsgcChart**: un descendiente de `Control` de Avalonia que pinta el gráfico asignado a su propiedad **Chart**, en Windows, Linux y macOS. Es la contrapartida en .NET del TsgcFMXChart de Delphi. El control registra un oyente en el gráfico mientras está conectado a un árbol visual y se repinta cada vez que el modelo cambia, dibujado con SkiaSharp en un mapa de bits de fotograma persistente y mostrado a través del préstamo de lienzo Skia del renderizador de Avalonia. El propio modelo no contiene código de dibujo, así que el mismo gráfico también puede mostrarse con una vista WinForms o WPF, o renderizarse sin interfaz visual, al mismo tiempo.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este control. Asignarlo registra el control como oyente mientras está conectado a una ventana; borrarlo, o liberar el gráfico, hace que el control deje de pintar nada.
FrameCount	uint (solo lectura)	Número de fotogramas completos renderizados desde que se creó el control.
OverlayFrameCount	uint (solo lectura)	Número de fotogramas solo de superposición renderizados desde que se creó el control (un cambio de retícula, información sobre herramientas o banda elástica que reutiliza el último contenido en caché en lugar de un repintado completo).

Las propiedades habituales de `Control` de Avalonia también están disponibles sin cambios: **Width**, **Height**, **Margin**, **HorizontalAlignment**, **VerticalAlignment**, **Focusable**, y los eventos de puntero, rueda y teclado.

Métodos

Método	Devuelve	Descripción
SaveToPNG(string aFileName)		Renderiza **Chart** al tamaño y DPI actuales del control en un nuevo archivo PNG. No hace nada cuando **Chart** es null.
SaveToSVG(string aFileName)		Renderiza **Chart** al tamaño y DPI actuales del control en un archivo SVG. No hace nada cuando **Chart** es null.

El ritmo de fotogramas sigue a **Chart.MaxFPS** igual que las demás vistas visuales: una notificación de cambio solo marca el control como sucio y programa un repintado, como máximo **Chart.MaxFPS** veces por segundo, de modo que un hilo en segundo plano puede alimentar las series del gráfico sin inundar el hilo de interfaz con repintados.

Ejemplo

Una aplicación Avalonia mínima que muestra un **TsgcAvaloniaChart** vinculado a un **TsgcChart** con una serie de línea, construida enteramente en código (sin necesidad de marcado XAML). Además de **esegece.sgcCharts.Avalonia**, el proyecto también necesita los paquetes NuGet **Avalonia.Desktop** y **Avalonia.Themes.Fluent**, igual que cualquier otra aplicación de escritorio Avalonia.

Véase también

- Información general de .NET
- Inicio Rápido: WinForms
- Inicio Rápido: WPF
- TsgcFMXChart (Delphi)

Renderizado sin interfaz visual

Paquete: esegece.sgcCharts.Rendering (destinado a netstandard2.0, net472, net6.0, net8.0 y net9.0)
TsgcChartRenderer es la vista sin interfaz visual de un modelo **TsgcChart**: una clase simple sin dependencia de ningún framework de interfaz y sin ventana propia, de modo que renderiza un gráfico en un archivo o flujo PNG, un documento SVG o un búfer de píxeles ARGB directo desde un programa de consola, un servicio o una aplicación ASP.NET. Funciona en Windows, Linux y macOS, incluido un contenedor Linux o un trabajo de CI sin pantalla conectada, y es la contrapartida en .NET del TsgcChartRenderer de Delphi. Al no tener superficie visual, no se ve afectado por **Chart.MaxFPS**: no hay bucle de fotogramas ni temporizador, cada llamada de render pinta el gráfico de forma síncrona en el momento en que se invoca.

Propiedades

Propiedad	Tipo	Descripción
Chart	TsgcChart	El modelo de gráfico que pinta este renderizador. Cada método Render* lanza **EsgcChartRendererError** cuando esto es null.
Width	int	Ancho lógico predeterminado en píxeles usado por las sobrecargas que no reciben tamaño. Valor predeterminado 800.
Height	int	Alto lógico predeterminado en píxeles usado por las sobrecargas que no reciben tamaño. Valor predeterminado 500.
Scale	double	Píxeles de dispositivo por píxel lógico. Valor predeterminado 1. La salida es Round(Width * Scale) por Round(Height * Scale) píxeles de dispositivo; **Chart.Scale** se establece en este valor durante el render y se restaura a su valor anterior al finalizar.
CompressPNG	bool	Indica si la salida PNG se comprime con deflate a través del codificador PNG de Skia. Valor predeterminado true.

Métodos

Método	Devuelve	Descripción
RenderToPNGFile(string aFileName) RenderToPNGFile(string aFileName, int aWidth, int aHeight)		Renderiza a un archivo PNG, con **Width**/**Height** o con el tamaño lógico indicado sin modificar **Width** y **Height**.
RenderToPNGStream(Stream aStream) RenderToPNGStream(Stream aStream, int aWidth, int aHeight)		Renderiza PNG (RGBA, 8 bits) a un flujo.
RenderToSVG() RenderToSVG(int aWidth, int aHeight)	string	Renderiza y devuelve un documento SVG 1.1 (ASCII puro) como cadena.
RenderToSVGFile(string aFileName) RenderToSVGFile(string aFileName, int aWidth, int aHeight)		Renderiza SVG a un archivo.

RenderToSVGStream(Stream aStream) RenderToSVGStream(Stream aStream, int aWidth, int aHeight)		Renderiza SVG a un flujo.
RenderToBuffer(ref uint[] aBuffer, out int aPixelWidth, out int aPixelHeight) RenderToBuffer(ref uint[] aBuffer, out int aPixelWidth, out int aPixelHeight, int aWidth, int aHeight)		Renderiza en píxeles 0xAARRGGBB directos (no premultiplicados), de arriba abajo, con un stride de fila igual a **aPixelWidth**; **aPixelWidth** y **aPixelHeight** se devuelven con el tamaño de dispositivo.

Concurrencia

Todo método de render puede llamarse desde cualquier hilo. Cada **TsgcChartRenderer** posee sus propios lienzos, uno rasterizado y uno SVG, protegidos por un bloqueo privado de esa instancia del renderizador. Además, todo render realizado por cualquier **TsgcChartRenderer** del proceso pasa por otro bloqueo compartido por todo el proceso, porque **TsgcChart.Render** mantiene el estado de diseño en el propio gráfico mientras se ejecuta. Mientras un render está en curso, el gráfico no debe modificarse estructuralmente, es decir, no se deben añadir, quitar ni liberar series o ejes, ni cambiar ninguna propiedad que afecte al diseño; añadir datos a través de los búferes de las series es seguro, porque las series dibujan a partir de una instantánea tomada bajo el propio bloqueo del búfer. Una vista visual (**TsgcAvaloniaChart**, una vista WinForms o WPF) no toma el bloqueo de render compartido, así que no se debe renderizar el mismo gráfico con un renderizador en un hilo mientras una vista visual lo está pintando en el mismo instante.

Ejemplo: render sin interfaz visual a PNG en Linux

Una aplicación de consola sin referencia a ningún framework de interfaz, compilable y ejecutable en Linux, que renderiza un gráfico a un archivo PNG.
Esto se ejecuta sin cambios en un contenedor Linux o un trabajo de CI: **esegece.sgcCharts.Rendering** incluye sus propios recursos nativos de SkiaSharp para Linux, así que no se requiere ningún servidor X, GPU ni instalación de fontconfig.

Ejemplo: pie y treemap renderizados sin interfaz visual

Las series pie, radar, heatmap, treemap y funnel (véase Series Gallery) se renderizan sin interfaz visual como cualquier otra serie. Un renderizador no dibuja superposición al pasar el ratón, por lo que la salida muestra solo el estado que guarda el modelo: establézcalo en código, como lo dejaría un clic en una vista, con **Exploded** en un sector de pie o **DrillPath** en un treemap. Un renderizador pinta varios gráficos uno tras otro cuando cambia **Chart**:
El SVG de estas series es idéntico, byte por byte, al SVG que escribe el **TsgcChartRenderer** de Delphi para el mismo gráfico y tamaño. Un heatmap cuyo **ImagePath** es true rellena sus celdas como una sola imagen, que el SVG incrusta como PNG. La demo **SeriesHeadless** de `demos\Charts` renderiza de este modo cada gráfico de la galería de series: `SeriesHeadless [folder] [-dark] [-width N] [-height N] [-scale N]`.

Véase también

- Información general de .NET
- Inicio Rápido: WinForms
- Series Gallery
- TsgcChartRenderer (Delphi)

Series Gallery

Paquete: esegece.sgcCharts (destinado a netstandard2.0, net472, net6.0, net8.0 y net9.0)
Las series pie, radar, heatmap, treemap y funnel son los equivalentes .NET de las series Delphi del mismo nombre. Viven en el paquete principal **esegece.sgcCharts**, por lo que se dibujan en todas las vistas (**TsgcWinFormsChart**, **TsgcWpfChart** y **TsgcAvaloniaChart**) y sin interfaz visual mediante **TsgcChartRenderer** (véase Headless Rendering), con las mismas propiedades, valores por defecto y diseño que en Delphi. El tema Delphi de cada serie, enlazado más abajo, describe cada propiedad. Esta página muestra cómo usarlas desde C# y enumera los puntos en los que la forma .NET de la API difiere.

Series

Clase	Modo de trazado	Descripción
TsgcChartSeries_Pie	cpmCell	Un gráfico circular, o un donut cuando **InnerRadiusPercent** es mayor que 0. Los sectores por debajo de **OthersThreshold** por ciento se fusionan en un sector Others, **ExplodeOnClick** separa un sector con un clic, y las etiquetas van dentro o fuera con líneas guía. Elementos: **Slices**, un **TsgcChartPieSlice** cada uno.
TsgcChartSeries_Radar	cpmPolar	Un polígono por serie sobre radios compartidos por todas las series radar del gráfico, **Filled** o solo contorno, sobre una rejilla poligonal o circular (**GridShape**). Elementos: **Points**, uno por radio.
TsgcChartSeries_Heatmap	cpmXY	Una cuadrícula de celdas coloreadas según una escala de color: un espectrograma en vivo alimentado línea a línea con **AppendLine** (**Capacity** conserva las líneas más recientes), o una matriz fija con filas y columnas etiquetadas sobre ejes de categorías. Datos: **Data**, un **TsgcChartHeatmapData**.
TsgcChartSeries_Treemap	cpmCell	Rectángulos anidados construidos a partir de rutas como `Europe/Spain/Madrid` (**PathDelimiter**), con una cabecera por grupo (**Levels**), desglose al hacer clic (**DrillOnClick**) y una ruta de migas para volver atrás. Elementos: **Nodes**, uno por hoja.
TsgcChartSeries_Funnel	cpmCell	Una etapa por elemento en el orden de la colección, que se estrecha hasta un cuello (**NeckWidthPercent**, **NeckHeightPercent**). Las etapas tienen la misma altura, o una altura según el valor con **ProportionalHeight**. Elementos: **Stages**.

El modo de trazado decide qué series pueden compartir un gráfico: véase Plot Areas. Un gráfico puede contener un pie y un funnel lado a lado, cada uno en su propia área de celdas, mientras que un heatmap comparte el trazado XY y sus ejes con las demás series XY.

Elementos y valores en vivo

Miembro	Descripción
AddItem(string aText, double aValue, uint aColor = 0)	Pie, radar, treemap y funnel. Añade un elemento y devuelve su índice. Un color 0 toma el color de paleta del índice del elemento, de modo que un elemento conserva su color mientras cambian los valores.

SetValueAt(int aIndex, double aValue) DeleteItem(int aIndex)	Cambia o elimina un elemento. El orden de los elementos no cambia. Un índice fuera de rango lanza `ArgumentOutOfRangeException`.
ItemCount()	El número de elementos, un método como toda función Delphi sin parámetros del port.
Slices, Points, Nodes, Stages	La colección de elementos. Su indexador devuelve un **TsgcChartCategoryItem** (**Text**, **Value**, **Color**); conviértalo a **TsgcChartPieSlice** para acceder a **Exploded**.
BeginUpdate() EndUpdate()	Agrupa muchos cambios en un único repintado y un único diseño.
AppendLine(double[] aValues) AppendLine(double aKey, double[] aValues)	Heatmap. Añade una línea de celdas, con la clave **aKey** o con el contador de líneas en curso del almacén. Los valores son un array, no una lista `params`.
SetLineLabels(string[] aLabels) SetCellLabels(string[] aLabels)	Heatmap. Las etiquetas de las líneas y de las celdas en un eje de categorías. Una sobrecarga acepta un **TsgcChartStrings**, la forma .NET del `TStrings` de Delphi, por lo que un `null` a secas es ambiguo: conviértalo explícitamente.
Cells(int aLine, int aCell) SetCells(int aLine, int aCell, double aValue)	Heatmap. Lee y escribe una celda, la forma de método de la propiedad indexada de Delphi `Cells[Line, Cell]`.

Interacción

Las vistas reenvían el ratón a **TsgcChart.Interaction**, por lo que los tooltips al pasar el ratón, el explode y el drill funcionan igual en WinForms, WPF y Avalonia. Un clic en una serie llama primero a **TsgcChart.OnSeriesClick** y después la serie lo procesa.

Miembro	Descripción
TsgcChartSeries_Pie.ExplodeOnClick	Por defecto false. Con true, un clic en un sector alterna su indicador **Exploded**, o **OthersExploded** para el sector Others.
TsgcChartSeries_Treemap.DrillOnClick	Por defecto true. Un clic en un mosaico desciende un nivel, al grupo bajo el clic, y un clic en una miga vuelve a subir hasta ella. **DrillDown(string aPath)**, **DrillUp()**, **DrillReset()** y **DrillPath** hacen lo mismo desde el código.
TsgcChartSeries_Treemap.OnDrill	Un **TsgcChartNotifyEvent** (`object Sender`) que se llama cuando cambia **DrillPath**. Una propiedad delegado, por lo que asignar un manejador sustituye al anterior, como en Delphi.
TsgcChart.OnSeriesClick	Un **TsgcChartHitEvent** (`object Sender, TsgcChartHitInfo aHit`). Para estas series **aHit.SubIndex** es el índice del elemento (-2 para el sector Others de un pie), **aHit.Text** el texto del sector o de la etapa o la ruta del treemap, y **aHit.Percent** la parte del elemento en un pie, un funnel o un treemap.

La superposición al pasar el ratón (el resaltado y el tooltip) solo se dibuja en pantalla. **TsgcChartRenderer**, **SaveToPNG** y **SaveToSVG** renderizan únicamente el modelo, de modo que un sector separado o un treemap desglosado aparece en su salida, mientras que un sector bajo el ratón no.

Ejemplo: construir las cinco series

Una clase auxiliar que construye un gráfico por serie, los mismos gráficos que la demo **SeriesGallery** con menos datos:

Ejemplo: alojar los gráficos en un formulario WinForms

Un **TsgcWinFormsChart** por pestaña, un título que sigue los clics y la ruta de drill, y valores en vivo. Con WPF use **TsgcWpfChart**, con Avalonia **TsgcAvaloniaChart**: el código del gráfico no cambia.

Demos

Demo	Framework	Descripción
Series-Gallery	WinForms, net8.0-windows	El pie, el radar, un espectrograma de 30 líneas por segundo, un heatmap de matriz, el treemap y el funnel, un **TsgcWinFormsChart** por pestaña, con un interruptor de donut, opciones de radar relleno o circular y botones Up y Top para el treemap.
Series-GalleryAvalonia	Avalonia, net8.0	La misma galería en controles **TsgcAvaloniaChart**, en Windows, Linux y macOS.
Series-Headless	Console, net8.0	Los gráficos de la galería renderizados con **TsgcChartRenderer** a archivos PNG y SVG, sin ventana: `SeriesHeadless [folder] [-dark] [-width N] [-height N] [-scale N]` .

Las demos están en `demos\Charts`. Ambas galerías aceptan `-seconds N` como autoprueba: muestran cada pestaña, hacen clic en un sector del pie y en un mosaico del treemap mediante la interacción del gráfico, se cierran tras N segundos y devuelven el código de salida 0 cuando todas las comprobaciones han pasado.

Véase también

- .NET Overview
- .NET Differences from Delphi
- Headless Rendering
- Quick Start: WinForms
- Plot Areas
- Colour Scale

Bridge

El paquete **esegece.sgcCharts.sgcWebSockets** (destinado a `netstandard2.0`, `net8.0`, `net9.0`) conecta las fuentes de datos de sgcCharts con una conexión sgcWebSockets en vivo: mensajes WebSocket, publicaciones MQTT, Server-Sent Events y flujos de mercado de Binance. Es un paquete independiente del paquete básico **esegece.sgcCharts** y requiere sgcWebSockets para compilar; un proyecto que solo use las fuentes de datos básicas no necesita este paquete ni sgcWebSockets en absoluto.
De forma predeterminada, el paquete hace referencia directamente al proyecto del puerto .NET de sgcWebSockets; una compilación sin ese árbol de código fuente puede referenciar en su lugar el paquete NuGet **esegece.sgcWebSockets**. Este puente no se incluye en la edición Community.

MQTT: TsgcChartDataSource_MQTT

Representa las publicaciones MQTT recibidas por un **TsgcWSProtocol_MQTT_Client**. Mientras la fuente está **Active** y **Client** está asignado, suscribe su propio manejador a **Client.OnMQTTPublish**.

Propiedad	Tipo	Descripción
Client	TsgcWSProtocol_MQTT_Client	El cliente MQTT cuyas publicaciones se representan.
TopicFilter	string	Filtro de tema MQTT, con los comodines `+` (un nivel) y `#` (este nivel y todos los inferiores). Un filtro vacío representa cada publicación.
PayloadFormat	TsgcChartMQTTPayloadFormat (cpfJSON, cpfNumber)	**cpfJSON** lee la carga útil a través de **Mapping** (un objeto JSON o un array de registros). **cpfNumber** analiza la carga útil como un número simple para Y y usa la hora local actual para X.
Active	bool (oculta la propiedad base)	La fuente redeclara **Active** para que, al establecerla, también se actualice el manejador en **Client**, igual que hace la asignación de **Client**.

WebSocket: TsgcChartDataSource_WebSocket

Representa mensajes JSON recibidos por un **TsgcWebSocketClient**. Mientras la fuente está **Active** y **Client** está asignado, suscribe su propio manejador a **Client.OnMessage**. Cada mensaje de texto se lee como JSON a través de **Mapping**.

Propiedad	Tipo	Descripción
Client	TsgcWebSocketClient	El cliente WebSocket cuyos mensajes se representan.
MessageFilter	string	Una ruta JSON que debe existir en un mensaje para que se represente, por ejemplo `"k"` para conservar solo los mensajes de kline de Binance. Un filtro vacío representa todo mensaje que se analice como JSON.
Active	bool (oculta la propiedad base)	Redeclarada como en **TsgcChartDataSource_MQTT**, de modo que el enganche sigue el valor.

Server-Sent Events: TsgcChartDataSource_SSE

Representa Server-Sent Events de dos fuentes independientes, que pueden usarse juntas en la misma fuente de datos.

- **Client** es un **TsgcHTTP1Client**, igual que en Delphi. La aplicación inicia el flujo con el método bloqueante `GetSSE(URL)` del cliente, normalmente desde un hilo de trabajo. Mientras la fuente está **Active** y **Client** está asignado, suscribe su propio manejador a **Client.OnSSEMessage**, que entrega el texto del flujo en bruto a medida que llega. La fuente almacena en búfer los fragmentos, los divide en eventos en cada línea en blanco, une las líneas `data:` de un evento, y lee el resultado como JSON a través de **Mapping**.
- **SSEClient** es un **TsgcSSEClient**, el cliente gestionado al estilo EventSource. No tiene contrapartida en Delphi dentro de este puente. Su **OnEvent** entrega eventos ya analizados, así que la fuente lee el **Data** del evento como JSON a través de **Mapping** directamente, sin ningún análisis de fragmentos de por medio.

Propiedad	Tipo	Descripción
Client	TsgcHTTP1Client	Cliente SSE bloqueante. Véase más arriba.
SSEClient	TsgcSSEClient	Cliente EventSource gestionado, una adición de .NET. Véase más arriba.
Active	bool (oculta la propiedad base)	Redeclarada como en **TsgcChartDataSource_MQTT**. Tanto **Client** como **SSEClient** se enganchan o desenganchan de forma independiente cuando cambia la propiedad.

Un método **ResetStream** descarta un evento almacenado en búfer, aún incompleto, del analizador de fragmentos de **Client**, por ejemplo tras una reconexión.

Binance: TsgcChartDataSource_Binance

Se suscribe y representa un flujo de datos de mercado público de Binance a través de un **TsgcWSAPI_Binance**. Los flujos de mercado de Binance no tienen un evento tipado: llegan como JSON en bruto en el evento **OnMessage** del cliente WebSocket al que está conectada la API (**API.Client**). Mientras la fuente está **Active** y **API** está asignado, suscribe allí su propio manejador; el enganche sigue a **API.Client** cada vez que se establece **API** o **Active**, y también en **Subscribe**.

Propiedad	Tipo	Descripción
API	TsgcWSAPI_Binance	El componente de API de Binance cuyo flujo se representa.
Symbol	string	El símbolo de Binance, por ejemplo `"BTCUSDT"`. Se pasa a minúsculas automáticamente para la suscripción, ya que los nombres de flujo de Binance son solo en minúsculas. Los mensajes de otro símbolo se ignoran.
Stream	TsgcChartBinanceStream (cbsKLine, cbsTicker, cbsTrade)	Qué evento de Binance se representa: klines (velas), el ticker de 24 horas (último precio), u operaciones individuales.
Interval	TsgcWSBinanceChartIntervals	Intervalo de la kline, por ejemplo `bci1m`. Solo se usa para **cbsKLine**; los mensajes de otro intervalo se ignoran.
ClosedOnly	bool	Solo **cbsKLine**. Cuando es true, solo se representan las velas cerradas; en caso contrario, la vela aún en formación se actualiza en el mismo lugar con cada mensaje.
Aggregator	TsgcChartAggregator (edición financiera)	Solo **cbsTrade**. Cuando está asignado, cada operación va a **Aggregator.AddTrade** en lugar de al búfer de la serie: la operación en sí no se representa, y la serie OHLC del agregador dibuja las velas (volumen incluido). Liberar el agregador no borra esta propie-

		dad; un agregador liberado ignora las operaciones que sigue recibiendo.
Active	bool (oculta la propiedad base)	Redeclarada como en **TsgcChartDataSource_MQTT**.

Eventos

Evento	Tipo	Descripción
OnTrade	TsgcChartBinanceTradeEvent(object Sender, double aTime, double aPrice, double aQty)	Solo **cbsTrade**. Se dispara para cada operación de **Symbol**, con su hora UTC, precio y cantidad, antes de que la operación se añada al búfer o se entregue a **Aggregator**.

Métodos

Método	Devuelve	Descripción
Subscribe()	int	Se suscribe a **Stream** para **Symbol** (y **Interval** para las klines) llamando a `SubscribeKLine` , `SubscribeTicker` o `SubscribeTrades` de **API**. Devuelve el id que devuelve la API.
Unsubscribe()	int	La llamada de cancelación de suscripción correspondiente.

Eventos multidifusión

Toda fuente de datos del puente engancha un evento del cliente que en la biblioteca Delphi es un simple puntero a método (**OnMQTTPublish**, **OnMessage**, **OnSSEMessage**, **OnEvent**) pero aquí es un evento multidifusión de .NET. La fuente Delphi guarda el manejador que ya estuviera asignado, instala el propio y restaura el guardado al desengancharse. En cambio, una fuente de datos .NET añade su propio manejador a la lista de invocación con `+=` al engancharse, y elimina solo ese mismo manejador con `-=` al desengancharse; cualquier manejador de la aplicación ya suscrito, o suscrito después, sigue disparándose y la fuente nunca lo toca.
La diferencia práctica está en el orden de suscripción, no en la presencia: los manejadores se ejecutan en el orden en que se añadieron, así que un manejador de la aplicación suscrito al cliente antes de que se enganche la fuente de datos se ejecuta antes de que la fuente de datos represente el mensaje, mientras que en Delphi siempre se representa primero. El enganche solo ocurre en tiempo de ejecución, nunca en tiempo de diseño. No existe **FreeNotification** en .NET, así que liberar el cliente subyacente no borra la propiedad **Client** (o **API**, o **SSEClient**) de la fuente de datos; en su lugar, establézcala en null, o libere la fuente de datos. En **TsgcChartDataSource_SSE**, un manejador de la aplicación en **OnSSEMessage** aún puede establecer **Cancel**, que es compartido por toda la lista de invocación, así que cualquier manejador que lo ponga en true detiene el flujo también para todos los demás manejadores.
Los mensajes, publicaciones y eventos llegan todos en el hilo desde el que notifique el cliente subyacente. Las adiciones pasan por el bloqueo y la notificación con limitación de frecuencia de la base **TsgcChartDataSource**, así que cualquier modo de notificación es seguro de usar con una fuente de datos del puente.

Ejemplo: gráfico en vivo de Binance

Véase también

- Bridge Overview (Delphi)
- Información general de .NET

Diferencias de .NET respecto a Delphi

sgcCharts .NET conserva los nombres de clase, propiedades y comportamiento de Delphi siempre que ambos lenguajes lo permiten (véase Información general de .NET). Esta página enumera los lugares donde el puerto .NET se comporta genuinamente de forma distinta a la biblioteca Delphi, no solo donde un nombre o un tipo cambió a su equivalente natural en C#.

Eventos

Área	Delphi	.NET
Fuentes de datos del puente en vivo (**esegece.sgcCharts.sgcWebSockets**): **OnMQTTPublish**, **OnMessage** (cliente WebSocket y cliente Binance), **OnSSEMessage** / **OnEvent**	Un simple puntero a método. La fuente de datos guarda el manejador actual del cliente, instala el propio y restaura el guardado al detenerse.	Un auténtico `event` multidifusión de C# (`+=` / `-=`). La fuente de datos suscribe y cancela la suscripción de su propio manejador en lugar de guardar y restaurar uno; véase Bridge Overview para el mecanismo completo.
Modelo de gráfico básico: **TsgcChart.OnChange**, **TsgcChartDataSource.OnDataChanged** / **OnBeforeAppend**, **TsgcChart.OnCrosshairMove** / **OnSeriesClick** / **OnZoom**, y miembros similares en ejes, series y dibujos	Un simple puntero a método, sustituido por asignación.	Sin diferencia: siguen siendo una simple propiedad delegada `get; set;` , no un `event` de C#, así que asignar un nuevo manejador sigue sustituyendo al anterior exactamente igual que en Delphi. Se enumeran aquí solo para contrastar con la fila del puente de arriba, ya que ambos se llaman **OnX**.

Propiedades indexadas

Área	Delphi	.NET
TsgcChartAxes, **TsgcChartPanes**, **TsgcChartDrawings**, **TsgcChartPalette**, **TsgcChartJSONNode**	Una propiedad de array indexada, p. ej. `Axes[Index]` .	Un auténtico indexador de C#: `this[int Index]` .
TsgcChartBuffer (`X` , `Y`), **TsgcChartOHLCBuffer** (`X` , `Open` , `High` , `Low` , `Close` , `Volume`), `XValues` y `Values` de las series financieras	Una propiedad de array indexada por columna, p. ej. `Open[Index]` , y una bidimensional, `Values[Line, Index]` .	Un método normal que recibe el índice como parámetro: `Open(Index)` , `Values(Line, Index)` . Una clase de C# solo puede declarar una firma de indexador (`this[...]`), y estas clases necesitan varios accesores por índice, con nombre propio o con varios parámetros, en la misma clase, así que se convierten en métodos en su lugar.

Las series pie, radar, heatmap, treemap y funnel y sus clases auxiliares: **TsgcChartSeries_Radar.SpokeText**, **TsgcChartSeries_Funnel.DrawnStages**, **TsgcChartSeries_Heatmap.Cells**, **TsgcChartHeatmapData.Cells**, **TsgcChartColorScale.LUT**, **TsgcChartPieLayout.Slices**, **TsgcChartTreemapLayout.Nodes**, **Crumbs** y **DrawnNodes**	Una propiedad array indexada, p. ej. `SpokeText[Index]`, `DrawnStages[Index]`, y la de lectura y escritura `Cells[Line, Cell]`.	Un método: `SpokeText(Index)`, `DrawnStages(Index)` (una copia del registro de la etapa), `Cells(Line, Cell)` para leer y `SetCells(Line, Cell, Value)` para escribir. `DrawnStages` lanza `ArgumentOutOfRangeException` donde Delphi lanza `EListError`. Los métodos de diseño `Slices`, `Nodes`, `Crumbs` y `DrawnNodes` también la lanzan para un índice fuera de `0..Count - 1`, donde Delphi lee el array interno sin comprobación.

Series pie, radar, heatmap, treemap y funnel

Área	Delphi	.NET
SetValueAt y **DeleteItem** de las series pie, radar, treemap y funnel	Un índice fuera de rango lanza `ERangeError`.	Un índice fuera de rango lanza `ArgumentOutOfRangeException`.
TsgcChartSeries_Heatmap: **AppendLine**, **SetLineLabels**, **SetCellLabels**	Un parámetro array abierto (`AppendLine([1, 2, 3])`) y una sobrecarga `TStrings` para las etiquetas.	Un parámetro `double[]` o `string[]`, no una lista `params`, y una sobrecarga **TsgcChartStrings** para las etiquetas. Un `null` a secas es ambiguo entre las dos sobrecargas de etiquetas, por lo que hay que convertirlo.
Los elementos (**Slices**, **Points**, **Nodes**, **Stages**) y los **Levels** del treemap	Se editan en el Object Inspector y se almacenan en el formulario (DFM).	No hay streaming del diseñador de formularios: añada los elementos en código con **AddItem** y los niveles con **Levels.Add()**. Véase Series Gallery.
Salida raster (**RenderToPNGFile**, **RenderToBuffer**, **SaveToPNG**)	Dibujada por el canvas raster de sgcUICore.	Dibujada por SkiaSharp, por lo que los píxeles suavizados del borde pueden diferir ligeramente de Delphi. El diseño es el mismo y la salida SVG es igual al SVG de Delphi byte por byte.

Carga desde un DataSet o un DataReader

TsgcChartDataSource_DataSet carga desde un `System.Data.DataTable` o cualquier `IDataReader`, los equivalentes en .NET del `TDataSet` de Delphi. Un `DataTable` no tiene cursor, así que no hay registro actual que guardar y restaurar tras la carga: la gestión de marcadores de la fuente de datos Delphi, que devuelve el `TDataSet` de origen al registro en el que estaba antes de que se ejecutara **Refresh**, no tiene contrapartida en .NET.

Ediciones

sgcCharts Delphi ofrece las ediciones Community, Trial y Registered. sgcCharts .NET solo ofrece **Community** (compilada con `-p:SgcCommunity=true`) y **Registered** (la compilación predeterminada): no hay compilación ni constante de prueba en .NET.

Véase también

- Información general de .NET
- Series Gallery
- Bridge Overview
- TsgcChartDataSource_DataSet

Solución de problemas

No hay problemas conocidos documentados todavía.

Introducción

sgcMaps es una biblioteca de componentes Delphi para mapas interactivos. Los controles de mapa dibujan teselas raster y vectoriales de forma nativa, sin control de navegador y sin JavaScript, y muestran encima marcadores, rutas, formas, etiquetas y mapas de calor. Los componentes no visuales añaden geocodificación, indicaciones de ruta, búsqueda de lugares, elevación, zonas horarias y ubicación del dispositivo, y el enrutador sin conexión y el geocodificador sin conexión responden sin ninguna conexión de red.
Un control VCL, un control FireMonkey y un control Lazarus (LCL) comparten un mismo modelo de mapa, de modo que el código que añade una capa, un marcador o una ruta es el mismo en todos los frameworks. El control de Lazarus no puede descargar teselas, así que en Lazarus muestra únicamente teselas de una carpeta. Un control de mapa también puede renderizar sin ventana, lo que convierte una aplicación de consola o un servicio en un renderizador de mapas estáticos.

- Descripción general: qué incluye sgcMaps y qué versión de Delphi necesita cada característica.
- Instalación: el instalador de sgcUI, los paquetes y los requisitos.
- Inicio rápido VCL e Inicio rápido FMX: un mapa funcional en pocas líneas.
- Controles de mapa: TsgcVCLMap y TsgcFMXMap, sus capas y superposiciones.
- Servicios en línea: geocodificación, indicaciones de ruta, lugares, elevación y zonas horarias.
- Rutas: edición interactiva de rutas y el enrutador sin conexión.
- Geocodificación sin conexión: búsqueda de lugares a partir de un índice local de GeoNames.
- Ubicación: la posición del dispositivo, real o simulada.
- .NET: sgcMaps for .NET, el puerto en C# con un control Windows Forms, renderizado sin interfaz visual y los servicios sin conexión.
- Demos, Herramientas, Lazarus y Solución de problemas: los proyectos de ejemplo, las herramientas de datos, el control LCL y los errores más habituales.

Descripción general

sgcMaps se construye en torno a un **modelo de mapa**: un conjunto de colecciones que describen lo que muestra un mapa. Layers contiene los orígenes de teselas, y Markers, Polylines, Polygons, Rectangles, Circles, Labels y Heatmaps contienen las superposiciones. Options y MarkerOptions definen el comportamiento de la vista. Las clases del modelo residen en **sgcMaps_Classes** y son las mismas en TsgcVCLMap y en TsgcFMXMap.
Los componentes no visuales trabajan con esas colecciones y no con un control concreto. TsgcMapRouteCalculator guarda sus puntos de paso en los Markers de un mapa y su ruta en sus Polylines, y TsgcMapLocation mantiene actualizados un marcador de posición y un círculo de precisión. Por eso el mismo componente sirve tanto para un formulario VCL como para un formulario FireMonkey.

Características

Teselas y capas

- Teselas raster en línea de proveedores integrados: OpenStreetMap, OpenTopoMap, Esri World Imagery, Esri World Street, CARTO Light y Dark, MapTiler Streets y Satellite, Thunderforest Cycle y Transport, y Stadia Alidade Smooth y Smooth Dark.
- Cualquier otro servidor de teselas mediante una plantilla de URL con marcadores de posición para {z}, {x}, {y}, subdominio, retina y clave de API.
- Una caché en disco por capa, de modo que las teselas ya vistas se dibujan sin conexión.
- Teselas sin conexión desde una carpeta z/x/y o desde un archivo MBTiles.
- Teselas vectoriales (Mapbox Vector Tiles) desde un servidor o desde un archivo MBTiles local, dibujadas con un estilo MapLibre o Mapbox GL.
- Varias capas con opacidad, rango de zoom, función de capa base o de superposición, y un selector de capas opcional en el mapa.
- Un descargador de teselas que llena una carpeta con todas las teselas de un área y de un rango de zoom, para crear un paquete de mapa sin conexión.

Superposiciones

- Marcadores con texto de ventana emergente y de información sobre herramientas, color, iconos de mapa de bits personalizados, anclajes y arrastre.
- Agrupación de marcadores, para que miles de marcadores sigan siendo legibles y rápidos.
- Polilíneas (continuas, discontinuas o geodésicas), polígonos con huecos, rectángulos y círculos, todos editables con el ratón.
- Etiquetas de texto y mapas de calor ponderados con varios degradados de color.

Interacción y salida

- Zoom con arrastre, rueda y botones, rotación del mapa (rumbo), una barra de escala y la atribución de las teselas.
- Eventos para clics, arrastres de marcadores, ediciones de formas, clics en agrupaciones, cambios de capa, cambios de zoom y errores de teselas.
- Renderizado a un mapa de bits, a un PNG o a un archivo BMP, también desde un control que no tiene ventana padre.

Datos

- Importación y exportación GeoJSON, GPX y KML de marcadores, polilíneas y polígonos.
- Funciones de geometría: distancia, rumbo, punto de destino, interpolación geodésica, punto en polígono, simplificación de líneas y punto más cercano de una polilínea.
- Un lector en streaming de archivos .osm.pbf y .osm XML de OpenStreetMap.

Servicios

- Geocodificación y geocodificación inversa con Nominatim, Google, HERE, Azure Maps, Mapbox, OpenRouteService y Geoapify.
- Indicaciones de ruta con OSRM, OpenRouteService, Google Routes, HERE, Azure Maps, Mapbox y Geoapify.
- Autocompletado de lugares con Photon, Nominatim, Google, HERE, Mapbox y Geoapify.
- Elevación de puntos y trayectos con Open-Elevation, OpenTopoData y Google.
- Zona horaria de una coordenada con TimeAPI, GeoNames, Google y Azure Maps.

Rutas, datos sin conexión y ubicación

- Edición interactiva de rutas con puntos de paso arrastrables y rutas alternativas.
- Enrutamiento sin conexión para coche, bicicleta y a pie sobre un grafo creado a partir de datos de OpenStreetMap.
- Geocodificación sin conexión sobre un índice SQLite creado a partir de los volcados de GeoNames.
- Ubicación del dispositivo en Windows, Android, iOS y macOS, además de simulación de posición y de recorrido para las pruebas.

Versiones de Delphi

El modelo de mapa, el control VCL, GPX, KML y las funciones de geometría compilan desde Delphi 7 hasta Delphi 13, y los paquetes cubren también de C++Builder 2007 a C++Builder 13. Algunas características dependen de partes de la RTL que llegaron más tarde, por lo que sus componentes se registran solo en las versiones del IDE que las tienen.

Característica	Unidad	Versión mínima
Control de mapa VCL, superposiciones, GPX, KML, geometría	sgcMaps_VCL, sgcMaps_Classes	Delphi 7
Salida PNG	sgcMaps_VCL	Delphi 2009
Teselas desde un servidor, descargador de teselas	sgcMaps_TileSource_HTTP, sgcMaps_TileDownloader	Delphi 2009 (Indy 10; THTTPClient desde Delphi XE8)
Calculador de rutas, enrutador sin conexión, teselas vectoriales desde un servidor	sgcMaps_RouteCalculator, sgcMaps_OfflineRouter, sgcMaps_TileSource_MVT	Delphi XE2
Control de mapa FireMonkey, ubicación del dispositivo, GeoJSON	sgcMaps_FMX, sgcMaps_Location, sgcMaps_GeoJSON	Delphi XE6
MBTiles, MBTiles vectorial, geocodificador sin conexión	sgcMaps_TileSource_MBTiles, sgcMaps_OfflineGeocoder	Delphi XE7 (FireDAC SQLite)
Servicios en línea	sgcMaps_Geocoding, sgcMaps_Directions, sgcMaps_Places, sgcMaps_Elevation, sgcMaps_TimeZone	Delphi XE8 (THTTPClient, System.JSON)

En Lazarus, el paquete contiene el control de mapa LCL, el origen de teselas de carpeta, los proveedores, el motor raster y las funciones de geometría. El control LCL no puede descargar teselas, así que muestra únicamente teselas de una carpeta, y GeoJSON, GPX y KML no se compilan bajo FPC. Consulte Lazarus.

.NET

sgcMaps también tiene un puerto .NET, sgcMaps for .NET, para .NET 8 y .NET 9. Tiene el mismo modelo de mapa, los mismos servicios en línea y los mismos archivos de datos sin conexión, un control Windows Forms y un renderizador sin interfaz visual que también funciona en Linux y macOS. Consulte la información general de .NET.

Instalación

sgcMaps se distribuye en el **instalador unificado de sgcUI**, junto con los demás productos visuales de eSeGeCe. El instalador sabe qué productos incluye su licencia, así que no hay nada que elegir a mano aparte de los IDE.

Instalación con el instalador de sgcUI

1. Cierre todas las copias abiertas del IDE e inicie el instalador de sgcUI.
2. Inicie sesión con su cuenta de eSeGeCe.
3. El instalador lee su licencia y muestra los productos que incluye. sgcMaps aparece marcado cuando su licencia lo cubre.
4. Seleccione las versiones de Delphi y de C++Builder en las que instalar e inicie la instalación.
5. Abra el IDE. Los componentes están en la página **SGC Maps** de la paleta de herramientas.

Los controles de mapa se muestran únicamente en la paleta de su propio framework: TsgcVCLMap en los proyectos VCL y TsgcFMXMap en los proyectos FireMonkey. Los componentes no visuales se muestran en ambas.

Requisitos

- A partir de Delphi XE7, el paquete de runtime requiere **FireDAC** con su controlador **SQLite** (los paquetes FireDAC, FireDACCommon, FireDACCommonDriver y FireDACSqliteDriver). sgcMaps lo utiliza para leer archivos MBTiles y el índice del geocodificador sin conexión, por lo que su edición de Delphi debe incluir FireDAC SQLite.
- Los servicios en línea usan THTTPClient (System.Net.HttpClient) y System.JSON, disponibles a partir de Delphi XE8.
- TsgcMapLocation usa System.Sensors, disponible a partir de Delphi XE6.

Paquetes

La carpeta **Packages** contiene un par de paquetes por cada versión del IDE: el paquete de runtime **sgcMapsD*nn*.dpk** y el paquete de tiempo de diseño **dclsgcMapsD*nn*.dpk**, desde Delphi 7 (D7) hasta Delphi 13 (D13). C++Builder usa los proyectos equivalentes **sgcMapsC*nn*.cbproj** y **dclsgcMapsC*nn*.cbproj**, desde C++Builder 2007 hasta C++Builder 13. Para instalar a mano, compile primero el paquete de runtime, después instale el paquete de tiempo de diseño y añada la carpeta Source a la ruta de biblioteca del IDE.

Lazarus

Lazarus tiene su propio paquete, **Packages\Lazarus\sgcMapsLazarus.lpk**. Ábralo con Package, Open Package File, compílelo y elija Use, Install, lo que reconstruye el IDE. Requiere los paquetes LCL, FCL e IDEIntf y contiene el control de mapa (TsgcVCLMap en la LCL), el origen de teselas de carpeta, los proveedores, el motor raster y las funciones de geometría. El control LCL no puede descargar teselas, así que muestra únicamente teselas de una carpeta. GeoJSON, GPX y KML figuran en el paquete pero no se compilan bajo FPC, y los servicios en línea, el calculador de rutas, el enrutador sin conexión, el geocodificador sin conexión y TsgcMapLocation están disponibles solo en Delphi. El tema Lazarus ofrece los detalles y un ejemplo.

Unidades

Añada a la cláusula uses la unidad de cada componente que cree en código. Los componentes colocados en un formulario añaden sus unidades automáticamente.

Unidad	Contenido
sgcMaps_Classes	El modelo de mapa: capas, marcadores, polilíneas, polígonos, rectángulos, círculos, etiquetas, mapas de calor y opciones, además de TsgcMapCoord.

sgcMaps_VCL	TsgcVCLMap.
sgcMaps_FMX	TsgcFMXMap.
sgcMaps_Providers	Los proveedores de teselas integrados y sgcMapApplyProvider.
sgcMaps_Services	TsgcMapService, la clase base de los servicios en línea.
sgcMaps_Geocoding	TsgcMapGeocoder y las clases de resultado de geocodificación.
sgcMaps_Directions	TsgcMapDirections, las clases de resultado de ruta y sgcMapAddRouteToPolylines.
sgcMaps_Places	TsgcMapPlaces.
sgcMaps_Elevation	TsgcMapElevation.
sgcMaps_TimeZone	TsgcMapTimeZone.
sgcMaps_RouteCalculator	TsgcMapRouteCalculator.
sgcMaps_OfflineRouter	TsgcMapOfflineRouter y TsgcMapRouteGraphBuilder.
sgcMaps_OfflineGeocoder	TsgcMapOfflineGeocoder y TsgcMapGeoIndexBuilder.
sgcMaps_Location	TsgcMapLocation.

.NET

El instalador de sgcUI no instala nada para .NET. sgcMaps for .NET se distribuye como sus propios paquetes NuGet (esegece.sgcMaps y sus paquetes complementarios), solo para la edición Registered. Consulte Instalación de .NET.

Inicio rápido VCL

Este tutorial muestra un mapa en un formulario VCL, primero con el diseñador de formularios y después enteramente en código.

1. En tiempo de diseño

1. Cree una aplicación VCL y coloque un **TsgcVCLMap** de la página SGC Maps en el formulario. Establezca su propiedad **Align** en alClient.
2. Añada la capa base en el evento **OnCreate** del formulario con **Layers.Add** y establezca su propiedad **Provider**. Una capa con proveedor descarga las teselas de ese proveedor sin ninguna otra configuración. La llamada opcional a **sgcMapApplyProvider** copia además en la capa el tipo de origen, la plantilla de URL, el zoom máximo y la atribución del proveedor.
3. Establezca la vista inicial con **SetCenter** y **Zoom**, y añada superposiciones con **AddMarker**, **AddPolyline** o **AddCircle**.
4. Ejecute la aplicación. Arrastre el mapa para desplazarlo, y use la rueda del ratón o los botones de zoom para acercar y alejar.

El manejador OnCreate completo para estos pasos es el ejemplo del tema TsgcVCLMap.

2. En código

El programa siguiente crea el mismo mapa sin archivo .dfm. El formulario crea su propio control, y por eso el constructor llama a **CreateNew** en lugar de al **Create** heredado. Un clic en el marcador abre su ventana emergente, y un clic en cualquier otro punto añade un marcador nuevo en esa posición.

```
program QuickStartVCL;

uses
  Forms, Classes, SysUtils, Controls,
  sgcMaps_Classes, sgcMaps_Providers, sgcMaps_VCL;

type
  TfrmMain = class(TForm)
  private
    FMap: TsgcVCLMap;
    procedure DoMarkerClick(Sender: TObject; aMarker: TsgcMapMarker);
    procedure DoMapClick(Sender: TObject; aLat, aLon: Double);
  public
    constructor Create(aOwner: TComponent); override;
  end;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oLayer: TsgcMapLayer;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcMaps Quick Start';
  Width := 900;
  Height := 600;

  // 1. the map control fills the form
  FMap := TsgcVCLMap.Create(Self);
  FMap.Parent := Self;
  FMap.Align := alClient;

  // 2. a base layer: OpenStreetMap raster tiles, cached on disk
  oLayer := FMap.Layers.Add;
  oLayer.Name := 'OpenStreetMap';
  oLayer.Provider := mtpOpenStreetMap;
  sgcMapApplyProvider(oLayer);
```

```
  oLayer.CacheFolder := ExtractFilePath(ParamStr(0)) + 'tile_cache';

  // 3. the initial view and a marker
  FMap.SetCenter(48.8584, 2.2945);
  FMap.Zoom := 14;
  FMap.AddMarker(48.8584, 2.2945, 'Eiffel Tower', 'Champ de Mars, Paris');

  // 4. events
  FMap.OnMarkerClick := DoMarkerClick;
  FMap.OnMapClick := DoMapClick;
end;

procedure TfrmMain.DoMarkerClick(Sender: TObject; aMarker: TsgcMapMarker);
begin
  FMap.ShowPopup(aMarker);
end;

procedure TfrmMain.DoMapClick(Sender: TObject; aLat, aLon: Double);
begin
  // every click on the map adds a marker at that position
  FMap.AddMarker(aLat, aLon, Format('%.5f, %.5f', [aLat, aLon]));
end;

var
  frmMain: TfrmMain;

begin
  Application.Initialize;
  Application.CreateForm(TfrmMain, frmMain);
  Application.Run;
end.
```

Acerca de las teselas

La capa guarda en caché en **CacheFolder** cada tesela que descarga, de modo que una vista ya vista se dibuja al instante y sin conexión. Los servidores de teselas de OpenStreetMap los mantienen voluntarios y tienen una política de uso que no permite un uso intensivo. Para una aplicación en producción, use un proveedor comercial como MapTiler, Thunderforest o Stadia (establezca el **ApiKey** de la capa), su propio servidor de teselas, o teselas sin conexión desde una carpeta o un archivo MBTiles. Mantenga habilitado **Options.ShowAttribution**, ya que los proveedores de teselas exigen que la atribución sea visible.

Siguientes pasos

- TsgcVCLMap: todas las propiedades, métodos y eventos del control.
- TsgcMapGeocoder: busque una dirección y muéstrela en el mapa.
- TsgcMapRouteCalculator: deje que el usuario cree una ruta haciendo clic en el mapa.

Inicio rápido FMX

TsgcFMXMap es el control de mapa de FireMonkey. Comparte el modelo de mapa con el control VCL, de modo que las capas, los marcadores y las formas se crean exactamente igual que en el Inicio rápido VCL. Necesita Delphi XE6 o posterior y funciona en las plataformas de escritorio y móviles de FireMonkey.

1. En tiempo de diseño

1. Cree una aplicación multidispositivo y coloque un **TsgcFMXMap** de la página SGC Maps en el formulario. Establezca su propiedad **Align** en Client.
2. Añada la capa base en el evento **OnCreate** del formulario con **Layers.Add** y establezca su propiedad **Provider**, lo que basta para descargar las teselas de ese proveedor. La llamada opcional a **sgcMapApplyProvider** copia además en la capa el tipo de origen, la plantilla de URL, el zoom máximo y la atribución del proveedor. En un dispositivo móvil la carpeta de la aplicación es de solo lectura, así que coloque la caché de teselas en una carpeta con permiso de escritura, como **TPath.GetCachePath**.
3. Establezca la vista inicial con **SetCenter** y **Zoom**, y añada marcadores.
4. Ejecute la aplicación. Desplácese con un dedo, acerque con el gesto de pellizco y gire con dos dedos cuando **Options.RotationEnabled** sea True.

El manejador OnCreate completo para estos pasos es el ejemplo del tema TsgcFMXMap.

2. En código

El programa siguiente crea el mapa en código. Activa la agrupación de marcadores, de modo que los marcadores cercanos se fusionan en un círculo numerado al alejar el mapa, y muestra la ventana emergente de un marcador cuando se toca.

```
program QuickStartFMX;

uses
  System.StartUpCopy, System.SysUtils, System.Classes, System.IOUtils,
  FMX.Forms, FMX.Types, FMX.Controls,
  sgcMaps_Classes, sgcMaps_Providers, sgcMaps_FMX;

type
  TfrmMain = class(TForm)
  private
    FMap: TsgcFMXMap;
    procedure DoMarkerClick(Sender: TObject; aMarker: TsgcMapMarker);
  public
    constructor Create(aOwner: TComponent); override;
  end;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oLayer: TsgcMapLayer;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcMaps FMX Quick Start';
  Width := 900;
  Height := 600;

  // 1. the map control fills the form
  FMap := TsgcFMXMap.Create(Self);
  FMap.Parent := Self;
  FMap.Align := TAlignLayout.Client;

  // 2. a base layer; the cache folder is valid on desktop and mobile
  oLayer := FMap.Layers.Add;
  oLayer.Name := 'OpenStreetMap';
  oLayer.Provider := mtpOpenStreetMap;
```

```
  sgcMapApplyProvider(oLayer);
  oLayer.CacheFolder := TPath.Combine(TPath.GetCachePath, 'tile_cache');

  // 3. the initial view, touch options and a few markers
  FMap.SetCenter(41.3874, 2.1686);
  FMap.Zoom := 13;
  FMap.Options.RotationEnabled := True;
  FMap.MarkerOptions.Clustering.Enabled := True;
  FMap.AddMarker(41.4036, 2.1744, 'Sagrada Familia');
  FMap.AddMarker(41.4145, 2.1527, 'Park Guell');
  FMap.AddMarker(41.3809, 2.1228, 'Camp Nou');

  // 4. events
  FMap.OnMarkerClick := DoMarkerClick;
end;

procedure TfrmMain.DoMarkerClick(Sender: TObject; aMarker: TsgcMapMarker);
begin
  FMap.ShowPopup(aMarker);
end;

var
  frmMain: TfrmMain;

begin
  Application.Initialize;
  Application.CreateForm(TfrmMain, frmMain);
  Application.Run;
end.
```

Siguientes pasos

- TsgcFMXMap: lo que difiere del control VCL.
- TsgcMapLocation: muestre y siga la posición del dispositivo.
- TsgcMapPlaces: sugiera lugares mientras el usuario escribe.

Demos

La carpeta **Demos\Maps** de la instalación de sgcUI contiene siete proyectos Delphi. Cada uno es una aplicación pequeña y completa que muestra un grupo de características, y entre todos utilizan todos los componentes de sgcMaps.

Demo	Framework	Versión mínima	Qué muestra
01 Mapa sin conexión	VCL	Delphi 7	Teselas de carpeta, de MBTiles y en línea, marcadores, formas, importación de GeoJSON, GPX y KML.
02 Características del mapa	VCL	Delphi 7	Proveedores y claves de API, agrupación, edición, mapas de calor, rotación y exportación.
03 Servicios	VCL	Delphi XE8	Búsqueda de lugares, geocodificación, indicaciones de ruta, elevación, zonas horarias y ubicación.
04 Servicios sin conexión	VCL	Delphi XE7	Teselas sin conexión, el geocodificador sin conexión y el enrutador sin conexión.
05 Renderizado estático	Consola	Delphi 7	Un mapa renderizado a un archivo PNG o BMP sin ventana.
06 FireMonkey	FireMonkey	Delphi XE6	El control FireMonkey en escritorio y móvil, con ubicación, búsqueda y rutas.
07 Teselas vectoriales	VCL	Delphi XE2	Teselas vectoriales desde un servidor o desde un archivo MBTiles local, con estilos, retina y rotación.

Las demos no llevan claves de API en su código fuente. Las que usan un proveedor que necesita clave tienen un cuadro de edición para introducirla, y todas las demos arrancan con servicios y teselas que no necesitan clave.
La mayoría de las demos descargan teselas de los servidores públicos de OpenStreetMap, lo que está bien para probar las demos pero no para una aplicación en producción. Consulte Solución de problemas para conocer las políticas de uso.

01 Mapa sin conexión

Carpeta	Proyecto	Frame-work	Versión mínima
Demos\Maps\01.OfflineMap	sgcMapsDemo.dpr	VCL	Delphi 7 (teselas en línea desde Delphi 2009, GeoJSON desde Delphi XE6, MB-Tiles desde Delphi XE7)

Qué muestra

Un TsgcVCLMap cuya capa base puede proceder de tres orígenes, elegidos con las opciones **Folder**, **MBTiles** y **Online**: una carpeta z/x/y de teselas, un archivo MBTiles o el servidor de teselas de OpenStreetMap con una caché en disco. El mapa arranca en Madrid con zoom 13.

- **Go To Coordinates** y un cuadro de zoom mueven la vista.
- **Marker at Center**, **Sample Markers** y **Clear Markers** añaden y quitan marcadores.
- **Sample Circle**, **Sample Polyline** y **Sample Polygon** dibujan formas.
- Las casillas **Zoom Controls** y **Scale Bar** activan y desactivan esas partes del mapa.
- **GeoJSON...**, **GPX...** y **KML...** importan un archivo en el mapa.
- El registro muestra los eventos OnMapClick, OnMarkerClick y OnZoomChanged.

Componentes y unidades

TsgcVCLMap con capas de carpeta, de MBTiles y HTTP, y las funciones sgcMapLoadGeoJSONFile, sgcMapLoadGPXFile y sgcMapLoadKMLFile. Unidades: sgcMaps_Classes, sgcMaps_TileSource, sgcMaps_VCL, sgcMaps_GeoJSON, sgcMaps_GPX y sgcMaps_KML.

Cómo ejecutarla

Abra **sgcMapsDemo.dpr** en el IDE, después compílelo y ejecútelo. La demo encuentra las unidades de sgcMaps a través de la ruta de biblioteca que establece el instalador de sgcUI; con una instalación manual, añada primero la carpeta Source a la ruta de biblioteca.

Datos y claves

- No hace falta ninguna clave de API.
- La carpeta **data** junto al proyecto contiene sample.geojson, sample.gpx y sample.kml para los botones de importación. La demo la busca primero junto al ejecutable y después en la carpeta actual.
- Una ruta de teselas relativa se resuelve desde la carpeta del ejecutable. La ruta predeterminada es **tiles**, una carpeta de archivos z\x\y.png que puede llenar con el descargador de teselas (véase Teselas sin conexión y el descargador de teselas).
- Las teselas en línea se guardan en caché en la carpeta **tile_cache** junto al ejecutable.
- Cuando la carpeta de teselas o el archivo MBTiles no existen, la demo escribe una línea en el registro.
- En las versiones de Delphi anteriores a XE6 el botón GeoJSON está desactivado, y en las anteriores a XE7 la opción MBTiles solo registra que necesita FireDAC.

02 Características del mapa

Carpeta	Proyecto	Framework	Versión mínima
Demos\Maps\02.MapFeatures	sgcMapsFeaturesDemo.dpr	VCL	Delphi 7 (teselas en línea y PNG desde Delphi 2009, GeoJSON desde Delphi XE6)

Qué muestra

Las características de dibujo e interacción de TsgcVCLMap, una pestaña para cada una:

Pestaña	Qué puede hacer
Layers	Elija una capa base, escriba una clave de API para MapTiler, Thunderforest o Stadia y pulse **Apply key** para añadir sus capas, y muestre el selector de capas en el mapa.
Clusters	Añada 2000 marcadores aleatorios alrededor de Madrid y active o desactive la agrupación.
Heatmap	Añada un mapa de calor de 5000 puntos ponderados y cambie su radio, su degradado y su opacidad.
Drawing	Añada un marcador arrastrable, una polilínea editable, un polígono editable con un hueco, un rectángulo editable y una línea geodésica de Madrid a Nueva York.
Rotation	Active la rotación, establezca el rumbo, gire con Ctrl+arrastre o con un arrastre con el botón derecho, y restablezca el mapa con el norte arriba.
Export	Exporte las superposiciones a GeoJSON, GPX o KML, y guarde el mapa como PNG (BMP antes de Delphi 2009).

El registro muestra OnMarkerDragEnd, OnShapeEdited, OnLayerChanged, OnClusterClick y OnBearingChanged.

Componentes y unidades

TsgcVCLMap, la lista de proveedores de sgcMaps_Providers (sgcMapGetProviderInfo), TsgcMapGPX, sgcMapSaveGeoJSONFile, sgcMapSaveKMLFile y sgcMapDistance. Unidades: sgcMaps_Classes, sgcMaps_VCL, sgcMaps_Providers, sgcMaps_Geometry, sgcMaps_GPX, sgcMaps_KML y sgcMaps_GeoJSON.

Cómo ejecutarla

Abra **sgcMapsFeaturesDemo.dpr** en el IDE, después compílelo y ejecútelo. La demo encuentra las unidades de sgcMaps a través de la ruta de biblioteca que establece el instalador de sgcUI; con una instalación manual, añada primero la carpeta Source a la ruta de biblioteca.

Datos y claves

- Al arrancar, la demo añade una capa base para cada proveedor que no necesita clave: OpenStreetMap, OpenTopoMap, las capas de Esri y CARTO Light y Dark.

- MapTiler, Thunderforest y Stadia necesitan una clave de API de sus sitios web. Escríbala en la pestaña Layers y pulse Apply key.
- Cada proveedor tiene su propia caché de teselas en **%TEMP%\sgcMapsFeaturesDemo**.
- Los archivos exportados se escriben donde elija en el cuadro de diálogo de guardado.

03 Servicios

Carpeta	Proyecto	Framework	Versión mínima
Demos\Maps\03.Services	sgcMapsServicesDemo.dpr	VCL	Delphi XE8

Qué muestra

Todos los servicios en línea en un solo formulario. Cada pestaña tiene una lista de proveedores y un cuadro para la clave de API, y el resultado se dibuja en el mapa.

Pestaña	Qué puede hacer
Search	Sugerencias de lugares mientras escribe (Photon, Nominatim, Google, HERE, Mapbox o Geoapify), y geocodificación de una dirección.
Reverse	Haga clic en el mapa para obtener la dirección de ese punto.
Directions	Una ruta entre dos direcciones o entre puntos de paso en los que se hace clic en el mapa (arrastrables), con un modo de viaje, alternativas, y OSRM, OpenRouteService, Google Routes, HERE, Azure Maps, Mapbox o Geoapify.
Elevation	El perfil de elevación de la ruta, con Open-Elevation, OpenTopoData o Google.
Time zone	La zona horaria en el centro del mapa, con TimeAPI, GeoNames, Google o Azure Maps.
Location	Simule un recorrido GPS a lo largo de la ruta, haga que el mapa siga la posición, o use la ubicación del dispositivo.

Componentes y unidades

TsgcMapPlaces, TsgcMapGeocoder, TsgcMapDirections, TsgcMapRouteCalculator, TsgcMapElevation, TsgcMapTimeZone y TsgcMapLocation, creados en código en un formulario con un TsgcVCLMap.

Cómo ejecutarla

Abra **sgcMapsServicesDemo.dpr** en el IDE, después compílelo y ejecútelo. La demo encuentra las unidades de sgcMaps a través de la ruta de biblioteca que establece el instalador de sgcUI; con una instalación manual, añada primero la carpeta Source a la ruta de biblioteca.

Datos y claves

- Los proveedores predeterminados no necesitan clave: Photon, Nominatim, el servidor de demostración de OSRM, OpenTopoData y TimeAPI.
- Google, HERE, Azure Maps, Mapbox, OpenRouteService y Geoapify necesitan una clave de sus consolas de desarrollador. Para GeoNames, escriba su nombre de usuario de GeoNames como clave.
- La demo establece su propio UserAgent en cada servicio, como piden los servidores públicos, y muestra sus límites en el formulario: una solicitud por segundo y ninguna solicitud masiva. En producción use su propio servidor o un proveedor comercial.
- Las teselas proceden de OpenStreetMap y se guardan en caché en la carpeta **tilecache** junto al ejecutable.

- Los errores de los servicios se escriben en el registro. Cuando la ubicación del dispositivo no está disponible, la demo lo indica y desmarca la opción.

04 Servicios sin conexión

Carpeta	Proyecto	Framework	Versión mínima
Demos\Maps\04.OfflineServices	sgcMapsOfflineDemo.dpr	VCL	Delphi XE7 (FireDAC SQLite)

Qué muestra

Búsqueda y cálculo de rutas sin conexión de red. La demo crea sus archivos de datos a partir de descargas gratuitas, los vuelve a abrir en el siguiente arranque y responde a cada solicitud de forma local. El mapa arranca en Andorra.

- **Offline tiles** con **Folder...** o **MBTiles...** establece un mapa base local. Sin él, o cuando falta el archivo, la demo usa las teselas en línea de OpenStreetMap y lo indica en el registro.
- **Build from GeoNames...** crea un índice de geocodificador a partir de un volcado de GeoNames, y **Open index...** abre uno existente.
- **Build from .osm.pbf...** crea un grafo de rutas a partir de un extracto de OpenStreetMap, y **Open graph...** abre uno existente.
- **Search** encuentra un lugar en el índice, y un clic en el mapa geocodifica de forma inversa ese punto.
- Los clics en el mapa añaden puntos de paso de la ruta, que se pueden arrastrar. La ruta se calcula para el perfil Car, Bike o Foot y se enumeran sus pasos.

Componentes y unidades

TsgcMapOfflineGeocoder, TsgcMapGeoIndexBuilder, TsgcMapOfflineRouter, TsgcMapRouteGraphBuilder y TsgcMapRouteCalculator, que llama al enrutador sin conexión desde su evento OnCalculateRoute.

Cómo ejecutarla

Abra **sgcMapsOfflineDemo.dpr** en el IDE, después compílelo y ejecútelo. La demo encuentra las unidades de sgcMaps a través de la ruta de biblioteca que establece el instalador de sgcUI; con una instalación manual, añada primero la carpeta Source a la ruta de biblioteca.

Datos y claves

- No hace falta ninguna clave de API.
- Un volcado de GeoNames de https://download.geonames.org/export/dump/, como ES.zip o cities15000.zip. Cuando admin1CodesASCII.txt y countryInfo.txt están en la misma carpeta, la demo también los utiliza. El índice se escribe junto al volcado con la extensión .sqlite.
- Un extracto de OpenStreetMap, como andorra-latest.osm.pbf de un servicio de descargas como Geofabrik. El grafo se escribe junto a él con la extensión .sgcgraph.
- Las compilaciones se ejecutan en un hilo con una barra de progreso, y el formulario no se puede cerrar hasta que terminan.
- Las rutas del índice, del grafo y de las teselas se guardan en un archivo .ini junto al ejecutable y se vuelven a abrir al arrancar.
- Las herramientas crean los mismos archivos desde la línea de comandos.

05 Renderizado estático

Carpeta	Proyecto	Frame-work	Versión mínima
Demos\Maps\05.StaticRender	sgcMapsStaticRender.dpr	Consola	Delphi 7 (teselas en línea y PNG desde Delphi 2009, rutas desde Delphi XE2, GeoJSON desde Delphi XE6)

Qué muestra

Una aplicación de consola que crea un TsgcVCLMap sin ventana padre, carga las teselas de una vista y la guarda como imagen. Es la base para un servicio de mapas estáticos o para imágenes de mapas en informes. Véase Renderizado de mapas estáticos.

Cómo ejecutarla

Compile el proyecto y ejecútelo desde un símbolo del sistema:

```
sgcMapsStaticRender --center lat,lon --zoom z --size WxH
  --provider osm|opentopo|esri|carto --out file.png|file.bmp
  [--marker lat,lon]... [--geojson file]
  [--route lat,lon;lat,lon[;...] --graph file.sgcgraph]
```

Parámetro	Significado
--center lat,lon	El centro de la vista. Obligatorio, salvo que se indique --route, que centra la vista en la ruta.
--zoom z	El nivel de zoom. El valor predeterminado es 13.
--size WxH	El tamaño de la imagen en píxeles. El valor predeterminado es 800x600.
--provider	osm (predeterminado), opentopo, esri o carto. Estos proveedores no necesitan clave.
--out file	La imagen que se escribe. Obligatorio. Una extensión .bmp escribe un archivo BMP, cualquier otra extensión un archivo PNG (un archivo BMP antes de Delphi 2009).
--marker lat,lon	Añade un marcador. Se puede repetir.
--geojson file	Dibuja las características de un archivo GeoJSON (Delphi XE6 y posteriores).
--route and --graph	Calcula una ruta a través de los puntos de paso con el enrutador sin conexión y el archivo .sgcgraph indicado, y la dibuja (Delphi XE2 y posteriores).

Las coordenadas usan un punto como separador decimal. El programa imprime el proveedor, la vista, el archivo guardado y la atribución, y termina con el código 0 si todo va bien, 1 si un parámetro es incorrecto y 2 cuando se produce una excepción.

```
sgcMapsStaticRender --center 48.8584,2.2945 --zoom 15 --size 1024x768 --provider osm --out eiffel.png
```

Datos y claves

- No hace falta ninguna clave de API.
- Las teselas se descargan cuando no están en la carpeta **tile_cache** junto al ejecutable, por lo que la primera ejecución necesita una conexión de red.
- La herramienta RouteGraphBuilder crea el archivo .sgcgraph para --route.

06 FireMonkey

Carpeta	Proyecto	Framework	Versión mínima
Demos\Maps\06.FMX	sgcMapsFMXDemo.dpr	FireMonkey	Delphi XE6 (búsqueda y rutas desde Delphi XE8)

Qué muestra

Un TsgcFMXMap creado en código, con cuatro pestañas:

Pestaña	Qué puede hacer
Search	Sugerencias de lugares mientras escribe, y **Route from here** hasta el lugar seleccionado.
Location	**Locate me** y **Follow my location**.
Over-lays	Agrupación de marcadores con 500 marcadores, un mapa de calor, un marcador arrastrable, un polígono editable y **Clear overlays**.
View	La capa base, el selector de capas, la rotación con dos dedos, el rumbo, **Reset compass** y **Snapshot to PNG**, que guarda sgcMapsFMX.png en la carpeta de documentos.

Componentes y unidades

TsgcFMXMap, TsgcMapLocation, TsgcMapPlaces con Photon y TsgcMapDirections con OSRM. El mapa se crea en código, por lo que la demo se ejecuta sin el paquete de tiempo de diseño.

Cómo ejecutarla

Abra **sgcMapsFMXDemo.dpr** en el IDE, elija la plataforma de destino, después compílelo y ejecútelo. En Android el permiso de ubicación se solicita en tiempo de ejecución (Delphi 10.3 y posteriores).

Datos y claves

- No hace falta ninguna clave de API. Las capas base son los proveedores que no necesitan clave, y Places y Directions establecen su propio UserAgent.
- La caché de teselas está en la carpeta temporal del dispositivo, que admite escritura en todas las plataformas.
- Cuando el dispositivo no tiene sensor de ubicación, cuando no llega ninguna posición en cinco segundos o cuando se deniega el permiso, la demo usa una posición simulada en Madrid para que el resto de la demo siga funcionando.

07 Teselas vectoriales

Carpeta	Proyecto	Framework	Versión mínima
Demos\Maps\07.VectorTiles	sgcMapsVectorDemo.dpr	VCL	Delphi XE2 (MBTiles local desde Delphi XE7)

Qué muestra

Teselas vectoriales dibujadas por TsgcVCLMap desde tres orígenes y con tres estilos. Véase Teselas vectoriales.

- Orígenes: **OpenFreeMap** (en línea, sin clave), **MapTiler vector** (necesita una clave) y un archivo **local .mbtiles** de teselas vectoriales.
- Estilos: el estilo integrado, el estilo de ejemplo **Liberty** o cualquier archivo JSON de estilo de MapLibre.
- **Retina** dibuja las teselas a 512 píxeles, y el control deslizante de rumbo o Ctrl+arrastre gira el mapa.
- La barra de estado muestra el zoom, el número de etiquetas dibujadas y los errores de teselas notificados por OnTileError.

Cómo ejecutarla

Abra **sgcMapsVectorDemo.dpr** en el IDE, después compílelo y ejecútelo. La demo encuentra las unidades de sgcMaps a través de la ruta de biblioteca que establece el instalador de sgcUI; con una instalación manual, añada primero la carpeta Source a la ruta de biblioteca.

Datos y claves

- OpenFreeMap no necesita clave. Al arrancar, la demo lee la URL actual de teselas del TileJSON de OpenFreeMap (Delphi XE8 y posteriores) y recurre a una URL fija cuando no puede.
- MapTiler necesita una clave del sitio web de MapTiler Cloud.
- Un archivo local debe contener teselas vectoriales con el esquema OpenMapTiles, el que dibuja el estilo integrado.
- El estilo de ejemplo Liberty se lee de los archivos de prueba de QA de la instalación. Cuando no se encuentra el archivo, esa opción se desactiva.

TsgcVCLMap

Unidad: sgcMaps_VCL
El control de mapa VCL. Desciende de TCustomControl y dibuja él mismo las teselas y las superposiciones, así que no necesita navegador, ni WebView, ni JavaScript. En Lazarus la misma clase es el control de mapa LCL, que muestra únicamente teselas de una carpeta porque no puede descargar teselas (véase Lazarus).
El control muestra las colecciones del modelo de mapa. Añada una capa de teselas a **Layers**, establezca la vista con **SetCenter** y **Zoom**, y añada marcadores y formas con los métodos Add o directamente a sus colecciones. El usuario se desplaza con el ratón, acerca o aleja con la rueda o con los botones de zoom, y gira el mapa con Ctrl+arrastre o con un arrastre con el botón derecho cuando **Options.RotationEnabled** es True.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
CenterLatitude	Double		Latitud del centro de la vista, en grados decimales (WGS 84).
CenterLongitude	Double		Longitud del centro de la vista, en grados decimales (WGS 84).
Zoom	Integer	13	Nivel de zoom, limitado por Options.MinZoom y Options.MaxZoom.
Layers	TsgcMapLayers		Capas de teselas, dibujadas en el orden de la colección. Véanse las propiedades de capa más abajo.
Markers	TsgcMapMarkers		Marcadores: Latitude, Longitude, PopupText, TooltipText, Color, IconBitmap, AnchorX, AnchorY, Draggable, Visible, Tag y Data.
Polylines	TsgcMapPolylines		Líneas: AddPoint, Color, Width, Opacity, Dashed, Geodesic, Editable, Visible y Tag.
Polygons	TsgcMapPolygons		Polígonos rellenos, con huecos.
Rectangles	TsgcMapRectangles		Rectángulos rellenos definidos por sus bordes norte, oeste, sur y este.
Circles	TsgcMapCircles		Círculos: CenterLat, CenterLon, RadiusMeters, FillColor, StrokeColor, StrokeWidth y FillOpacity.
Labels	TsgcMapLabels		Etiquetas de texto ancladas a una coordenada.
Heatmaps	TsgcMapHeatmaps		Mapas de calor de puntos ponderados, con los degradados mhgDefault, mhgFire, mhgCool y mhgGrayscale.
Options	TsgcMapOptions		Opciones de la vista: MinZoom, MaxZoom, TileSize, ShowZoomControls, ShowScaleBar, ShowAttribution, AttributionText, BackgroundColor, DragEnabled, ZoomEnabled, Bearing, RotationEnabled y ShowLayerSwitcher.

MarkerOptions	TsgcMapMarkerOptions		Tamaño, color y fuente predeterminados de los marcadores, y Clustering (Enabled, GridSize, MinClusterSize, Color, FontColor, ZoomOnClick, MaxZoom).
ActivePopupMarker	TsgcMapMarker		El marcador cuya ventana emergente está abierta, o nil. Solo lectura.
DrawnMarkerCount	Integer		Marcadores dibujados en el último repintado. Solo lectura.
DrawnClusterCount	Integer		Agrupaciones dibujadas en el último repintado. Solo lectura.
DrawnLabelCount	Integer		Etiquetas dibujadas en el último repintado. Solo lectura.
LastRenderMissingTiles	Integer		Teselas que aún no estaban disponibles en el último repintado. Solo lectura.

Propiedades de capa

Cada elemento de **Layers** es un TsgcMapLayer.

Propiedad	Tipo	Predeterminado	Descripción
Name	string		Nombre que muestra el selector de capas.
TileSourceType	TsgcMapTileSourceType	mtsFolder	mtsFolder (carpeta z/x/y), mtsMBTiles, mtsHTTP, mtsVectorMBTiles o mtsVectorHTTP.
TileSourcePath	string		Carpeta o archivo MBTiles de una capa sin conexión.
URLTemplate	string		URL de teselas de una capa en línea, con los marcadores de posición {z}, {x} y {y}.
CacheFolder	string		Caché en disco de una capa en línea. Vacío desactiva la caché.
Provider	TsgcMapTileProvider	mtpCustom	Un proveedor integrado. La capa descarga sus teselas sin ninguna otra configuración; sgcMapApplyProvider copia además en la capa el tipo de origen, la plantilla de URL, el zoom máximo y la atribución del proveedor.
ApiKey	string		Clave de los proveedores que la necesitan (MapTiler, Thunderforest, Stadia).
Attribution	string		Texto de atribución de la capa.
Opacity	Double		Opacidad de 0 a 1.
Visible	Boolean	True	Muestra u oculta la capa.

MinZoom, MaxZoom	Integer	0, 19	Rango de zoom de las teselas de la capa.
IsBaseLayer	Boolean	True	True para una capa base, False para una superposición que se dibuja encima. Mostrar una capa base con SetLayerVisible o en el selector de capas oculta las demás capas base.
Retina	Boolean	False	Solicita teselas de alta resolución.
VectorStyleFile	string		Estilo de MapLibre o de Mapbox GL de una capa vectorial.

Métodos

Método	Descripción
SetCenter(aLat, aLon: Double)	Mueve el centro de la vista.
ZoomIn, ZoomOut	Cambia el zoom en un nivel.
FitBounds(aNorthLat, aWestLon, aSouthLat, aEastLon: Double; aPaddingPx: Integer = 20)	Centra la vista y ajusta el zoom para mostrar un área.
FitToMarkers(aPaddingPx: Integer = 20)	Centra la vista y ajusta el zoom para mostrar todos los marcadores.
FitToPolyline(aPolyline: TsgcMapPolyline; aPaddingPx: Integer = 20)	Centra la vista y ajusta el zoom para mostrar una polilínea, por ejemplo una ruta.
AddMarker(aLat, aLon: Double; const aPopup: string = ''; const aTooltip: string = '')	Añade un marcador con un texto de ventana emergente y una información sobre herramientas opcionales.
ClearMarkers	Elimina todos los marcadores.
AddPolyline: TsgcMapPolyline	Añade una polilínea vacía; añada sus puntos con AddPoint.
AddPolygon: TsgcMapPolygon	Añade un polígono vacío.
AddCircle(aLat, aLon, aRadiusMeters: Double): TsgcMapCircle	Añade un círculo.
AddRectangle(aNorthLat, aWestLon, aSouthLat, aEastLon: Double): TsgcMapRectangle	Añade un rectángulo.
AddLabel(aLat, aLon: Double; const aText: string): TsgcMapLabel	Añade una etiqueta de texto.
ShowPopup(aMarker: TsgcMapMarker), HidePopup	Abre o cierra la ventana emergente de un marcador.
MarkerAtScreen(aX, aY): TsgcMapMarker	El marcador situado en un punto del control, o nil.
ClusterAtScreen(aX, aY; out aCount: Integer; out aLat, aLon: Double): Boolean	True cuando hay una agrupación en un punto del control, con su tamaño y su posición.

SetLayerVisible(aIndex: Integer; aVisible: Boolean)	Muestra u oculta una capa. Mostrar una capa base oculta las demás capas base, que es también lo que hace el selector de capas.
RefreshTiles	Recarga las teselas de la vista.
AttributionText: string	La atribución de las capas visibles.
RenderToBitmap(aBitmap: TBitmap; aIncludeControls: Boolean = False)	Dibuja la vista actual en un mapa de bits.
SaveToPNG, SaveToBMP(const aFileName: string; aWidth, aHeight: Integer; aIncludeControls: Boolean = False)	Renderiza la vista al tamaño indicado en un archivo PNG o BMP. Funciona sin ventana padre.

Eventos

Evento	Descripción
OnMapClick	Un clic en el mapa, con su latitud y su longitud.
OnMarkerClick	Un clic en un marcador.
OnMarkerDragStart, OnMarkerDrag, OnMarkerDragEnd	El usuario arrastra un marcador cuya propiedad Draggable es True.
OnShapeEdited	El usuario movió un vértice de una polilínea, un polígono o un rectángulo editables.
OnClusterClick	Un clic en una agrupación, con su tamaño y su posición. Establezca aHandled en True para gestionar el clic usted mismo.
OnLayerChanged	El usuario eligió otra capa en el selector de capas.
OnZoomChanged	Cambió el nivel de zoom.
OnBearingChanged	Se giró el mapa.
OnTileError	No se pudo cargar una tesela, con la capa, las coordenadas de la tesela y el error.

Ejemplo: configuración del mapa

Un manejador OnCreate que añade una capa de OpenStreetMap, centra la vista y añade un marcador, una línea y un círculo.

```
procedure TForm1.FormCreate(Sender: TObject);
var
  oLayer: TsgcMapLayer;
  oLine: TsgcMapPolyline;
  oCircle: TsgcMapCircle;
begin
  // base layer: OpenStreetMap tiles, cached on disk
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'OpenStreetMap';
  oLayer.Provider := mtpOpenStreetMap;
  sgcMapApplyProvider(oLayer);
  oLayer.CacheFolder := ExtractFilePath(ParamStr(0)) + 'tile_cache';
```

```
  // center on Madrid
  sgcVCLMap1.SetCenter(40.4168, -3.7038);
  sgcVCLMap1.Zoom := 12;

  // a marker with a popup text and a tooltip
  sgcVCLMap1.AddMarker(40.4153, -3.6844, 'Retiro Park', 'Click for details');

  // a polyline between two points
  oLine := sgcVCLMap1.AddPolyline;
  oLine.AddPoint(40.4168, -3.7038);
  oLine.AddPoint(40.4153, -3.6844);
  oLine.Width := 4;

  // a circle with a radius of 500 meters
  oCircle := sgcVCLMap1.AddCircle(40.4168, -3.7038, 500);
  oCircle.FillOpacity := 0.2;
end;
```

Ejemplo: eventos e instantáneas

```
procedure TForm1.sgcVCLMap1MarkerClick(Sender: TObject; aMarker: TsgcMapMarker);
begin
  sgcVCLMap1.ShowPopup(aMarker);
end;

procedure TForm1.sgcVCLMap1MapClick(Sender: TObject; aLat, aLon: Double);
begin
  memoLog.Lines.Add(Format('Clicked at %.5f, %.5f', [aLat, aLon]));
end;

procedure TForm1.btnSnapshotClick(Sender: TObject);
begin
  // renders the current view into an 800 x 600 PNG file
  sgcVCLMap1.SaveToPNG('map.png', 800, 600);
end;
```

Véase también

- TsgcFMXMap
- Inicio rápido VCL
- TsgcMapRouteCalculator

TsgcFMXMap

Unidad: sgcMaps_FMX
El control de mapa de FireMonkey, disponible desde Delphi XE6. Desciende del TControl de FireMonkey y comparte el modelo de mapa con TsgcVCLMap: las mismas Layers, Markers, Polylines, Polygons, Rectangles, Circles, Labels, Heatmaps, Options y MarkerOptions, los mismos métodos y los mismos eventos.

Diferencias respecto a TsgcVCLMap

- Entrada táctil: un dedo desplaza, el gesto de pellizco acerca o aleja y, con **Options.RotationEnabled**, dos dedos giran el mapa.
- **MarkerAtScreen** y **ClusterAtScreen** reciben coordenadas Single, el tipo de coordenada de FireMonkey.
- **RenderToBitmap** recibe un TBitmap de FMX, y el control publica **HitTest** (True de forma predeterminada).
- En las plataformas móviles la carpeta de la aplicación es de solo lectura: ponga el **CacheFolder** de la capa en una carpeta con permiso de escritura, como TPath.GetCachePath, y guarde las instantáneas en TPath.GetDocumentsPath.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
CenterLatitude	Double		Latitud del centro de la vista, en grados decimales (WGS 84).
CenterLongitude	Double		Longitud del centro de la vista, en grados decimales (WGS 84).
Zoom	Integer	13	Nivel de zoom, limitado por Options.MinZoom y Options.MaxZoom.
Layers	TsgcMapLayers		Capas de teselas, dibujadas en el orden de la colección. Véanse las propiedades de capa más abajo.
Markers, Polylines, Polygons, Rectangles, Circles, Labels, Heatmaps	collections		Las superposiciones, como en TsgcVCLMap.
Options, MarkerOptions	TsgcMapOptions, TsgcMapMarkerOptions		Opciones de vista, de marcadores y de agrupación, como en TsgcVCLMap.
Canvas	TCanvas		El lienzo de FireMonkey del control. Solo lectura.
ActivePopupMarker	TsgcMapMarker		El marcador cuya ventana emergente está abierta, o nil. Solo lectura.
DrawnMarkerCount	Integer		Marcadores dibujados en el último repintado. Solo lectura.
DrawnClusterCount	Integer		Agrupaciones dibujadas en el último repintado. Solo lectura.

DrawnLabelCount	Integer		Etiquetas dibujadas en el último repintado. Solo lectura.
LastRenderMissingTiles	Integer		Teselas que aún no estaban disponibles en el último repintado. Solo lectura.

Métodos

Método	Descripción
SetCenter(aLat, aLon: Double)	Mueve el centro de la vista.
ZoomIn, ZoomOut	Cambia el zoom en un nivel.
FitBounds(aNorthLat, aWestLon, aSouthLat, aEastLon: Double; aPaddingPx: Integer = 20)	Centra la vista y ajusta el zoom para mostrar un área.
FitToMarkers(aPaddingPx: Integer = 20)	Centra la vista y ajusta el zoom para mostrar todos los marcadores.
FitToPolyline(aPolyline: TsgcMapPolyline; aPaddingPx: Integer = 20)	Centra la vista y ajusta el zoom para mostrar una polilínea, por ejemplo una ruta.
AddMarker(aLat, aLon: Double; const aPopup: string = ''; const aTooltip: string = '')	Añade un marcador con un texto de ventana emergente y una información sobre herramientas opcionales.
ClearMarkers	Elimina todos los marcadores.
AddPolyline: TsgcMapPolyline	Añade una polilínea vacía; añada sus puntos con AddPoint.
AddPolygon: TsgcMapPolygon	Añade un polígono vacío.
AddCircle(aLat, aLon, aRadiusMeters: Double): TsgcMapCircle	Añade un círculo.
AddRectangle(aNorthLat, aWestLon, aSouthLat, aEastLon: Double): TsgcMapRectangle	Añade un rectángulo.
AddLabel(aLat, aLon: Double; const aText: string): TsgcMapLabel	Añade una etiqueta de texto.
ShowPopup(aMarker: TsgcMapMarker), HidePopup	Abre o cierra la ventana emergente de un marcador.
MarkerAtScreen(aX, aY): TsgcMapMarker	El marcador situado en un punto del control, o nil.
ClusterAtScreen(aX, aY; out aCount: Integer; out aLat, aLon: Double): Boolean	True cuando hay una agrupación en un punto del control, con su tamaño y su posición.
SetLayerVisible(aIndex: Integer; aVisible: Boolean)	Muestra u oculta una capa. Mostrar una capa base oculta las demás capas base, que es también lo que hace el selector de capas.
RefreshTiles	Recarga las teselas de la vista.
AttributionText: string	La atribución de las capas visibles.

RenderToBitmap(aBitmap: TBitmap; aIncludeControls: Boolean = False)	Dibuja la vista actual en un mapa de bits.
SaveToPNG, SaveToBMP(const aFileName: string; aWidth, aHeight: Integer; aIncludeControls: Boolean = False)	Renderiza la vista al tamaño indicado en un archivo PNG o BMP. Funciona sin ventana padre.

Eventos

Evento	Descripción
OnMapClick	Un clic en el mapa, con su latitud y su longitud.
OnMarkerClick	Un clic en un marcador.
OnMarkerDragStart, OnMarkerDrag, OnMarkerDragEnd	El usuario arrastra un marcador cuya propiedad Draggable es True.
OnShapeEdited	El usuario movió un vértice de una polilínea, un polígono o un rectángulo editables.
OnClusterClick	Un clic en una agrupación, con su tamaño y su posición. Establezca aHandled en True para gestionar el clic usted mismo.
OnLayerChanged	El usuario eligió otra capa en el selector de capas.
OnZoomChanged	Cambió el nivel de zoom.
OnBearingChanged	Se giró el mapa.
OnTileError	No se pudo cargar una tesela, con la capa, las coordenadas de la tesela y el error.

Ejemplo: configuración del mapa

Un manejador OnCreate que añade una capa de OpenStreetMap con su caché en la carpeta de caché de la aplicación, activa la rotación y la agrupación, y guarda una instantánea en la carpeta de documentos.

```
procedure TForm1.FormCreate(Sender: TObject);
var
  oLayer: TsgcMapLayer;
begin
  oLayer := sgcFMXMap1.Layers.Add;
  oLayer.Name := 'OpenStreetMap';
  oLayer.Provider := mtpOpenStreetMap;
  sgcMapApplyProvider(oLayer);
  // on mobile, keep the tile cache in the cache folder of the application
  oLayer.CacheFolder := TPath.Combine(TPath.GetCachePath, 'tile_cache');

  sgcFMXMap1.SetCenter(41.3874, 2.1686);
  sgcFMXMap1.Zoom := 13;
  // two finger rotation and marker clustering
  sgcFMXMap1.Options.RotationEnabled := True;
  sgcFMXMap1.MarkerOptions.Clustering.Enabled := True;
  sgcFMXMap1.AddMarker(41.4036, 2.1744, 'Sagrada Familia');
  sgcFMXMap1.OnMarkerClick := sgcFMXMap1MarkerClick;
end;

procedure TForm1.sgcFMXMap1MarkerClick(Sender: TObject; aMarker: TsgcMapMarker);
begin
```

```
  sgcFMXMap1.ShowPopup(aMarker);
end;

procedure TForm1.btnSnapshotClick(Sender: TObject);
begin
  sgcFMXMap1.SaveToPNG(TPath.Combine(TPath.GetDocumentsPath, 'map.png'),
    1024, 768);
end;
```

Véase también

- TsgcVCLMap
- Inicio rápido FMX
- TsgcMapLocation

Teselas, capas y proveedores

Un mapa se dibuja a partir de **teselas**: imágenes cuadradas, de 256 píxeles de ancho de forma predeterminada, que cubren el mundo en cada nivel de zoom. En el zoom 0 una tesela muestra el mundo entero, y cada nivel siguiente duplica el número de teselas en cada dirección. Una tesela se direcciona por su zoom, x e y, el esquema z/x/y que usan OpenStreetMap y casi todos los servidores de teselas.
La colección **Layers** de TsgcVCLMap y de TsgcFMXMap indica de dónde proceden las teselas. Cada elemento es un **TsgcMapLayer** (unidad sgcMaps_Classes), y su **TileSourceType** selecciona el origen:

TileSourceType	Las teselas proceden de	Ruta o URL
mtsFolder	Una carpeta z/x/y de archivos .png, .jpg o .jpeg en disco.	TileSourcePath
mtsMBTiles	Un archivo MBTiles raster.	TileSourcePath
mtsHTTP	Un servidor de teselas, mediante una plantilla de URL.	URLTemplate
mtsVectorHTTP	Un servidor de teselas vectoriales. Véase Teselas vectoriales.	URLTemplate
mtsVectorMBTiles	Un archivo MBTiles vectorial. Véase Teselas vectoriales.	TileSourcePath

Los orígenes de carpeta y de MBTiles funcionan sin conexión de red; Teselas sin conexión y el descargador de teselas explica cómo prepararlos.

Proveedores integrados

La propiedad **Provider** de una capa nombra uno de los servidores de teselas integrados de **TsgcMapTileProvider**. Llame a **sgcMapApplyProvider** (unidad sgcMaps_Providers) después de establecerla: establece TileSourceType en mtsHTTP, copia en la capa la plantilla de URL y el zoom máximo del proveedor, y rellena Attribution cuando está vacío. Una capa cuyo Provider no es mtpCustom descarga las teselas de su proveedor incluso sin esa llamada, pero entonces MaxZoom conserva su valor predeterminado de 19 y las propiedades de la capa no muestran la configuración del proveedor.

Proveedor	Clave de API	Zoom máximo	Teselas retina
mtpOpenStreetMap	No	19	No
mtpOpenTopoMap	No	17	No
mtpEsriWorldImagery	No	19	No
mtpEsriWorldStreet	No	19	No
mtpCartoLight	No	20	Sí
mtpCartoDark	No	20	Sí
mtpMapTilerStreets	Sí	22	Sí
mtpMapTilerSatellite	Sí	20	Sí
mtpThunderforestCycle	Sí	22	Sí
mtpThunderforestTransport	Sí	22	Sí
mtpStadiaAlidadeSmooth	Sí	20	Sí

mtpStadiaAlidadeSmoothDark	Sí	20	Sí

Establezca la clave de los proveedores que la necesitan en la propiedad **ApiKey** de la capa. Los servidores públicos tienen políticas de uso: tile.openstreetmap.org, por ejemplo, prohíbe el uso intensivo y las descargas masivas, así que una aplicación de producción debe usar un proveedor comercial o su propio servidor de teselas.

Plantillas de URL

Para cualquier otro servidor, establezca TileSourceType en mtsHTTP y escriba su **URLTemplate**. Estos marcadores de posición se sustituyen en cada tesela:

Marcador de posición	Se sustituye por
{z}, {x}, {y}	El zoom y la columna y la fila de la tesela.
{s}	Una letra de subdominio, rotada por tesela para que las solicitudes se repartan entre los servidores. Las letras son a, b y c salvo que el proveedor defina otras.
{r}	@2x cuando la capa tiene Retina establecido en True, vacío en caso contrario.
{apikey}	El ApiKey de la capa.

La misma expansión está disponible como **sgcMapExpandTileURL**. **sgcMapProviderNames** rellena una lista de cadenas con los nombres de los proveedores, y **sgcMapGetProviderInfo** devuelve un registro **TsgcMapProviderInfo** con el nombre, la plantilla de URL, los subdominios, la atribución, el zoom máximo y si el proveedor necesita una clave de API o admite teselas retina.

```
procedure TForm2.ListProviders;
var
  oInfo: TsgcMapProviderInfo;
begin
  // every provider name; Objects[i] holds Ord(provider)
  sgcMapProviderNames(cboProvider.Items);

  oInfo := sgcMapGetProviderInfo(mtpOpenTopoMap);
  memoLog.Lines.Add(Format('%s: max zoom %d, API key %s',
    [oInfo.Name, oInfo.MaxZoom, BoolToStr(oInfo.RequiresApiKey, True)]));
  // the URL of tile 10/523/380
  memoLog.Lines.Add(sgcMapExpandTileURL(oInfo.URLTemplate, oInfo.Subdomains,
    10, 523, 380, False, ''));
end;
```

Varias capas

Una capa es una **capa base** (IsBaseLayer True, el valor predeterminado) o una **superposición**. El mapa dibuja la primera capa base visible y después todas las superposiciones visibles encima de ella, en el orden de la colección. Opacity establece la transparencia de una capa, MinZoom y MaxZoom limitan los niveles de zoom en los que se dibuja, y Retina pide al proveedor teselas al doble de resolución.
Establezca **Options.ShowLayerSwitcher** en True para que el usuario pueda elegir las capas en el mapa. Las capas base se comportan como elementos de opción: mostrar una con **SetLayerVisible** oculta las demás capas base. El evento **OnLayerChanged** informa de la capa que se mostró o se ocultó.

```
procedure TForm2.SetupLayers;
var
  oLayer: TsgcMapLayer;
begin
  // base layer: a built-in provider
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'CARTO Light';
  oLayer.Provider := mtpCartoLight;
```

```
  sgcMapApplyProvider(oLayer);
  oLayer.Retina := True;
  oLayer.CacheFolder := ExtractFilePath(ParamStr(0)) + 'cache\carto';

  // second base layer, hidden until chosen in the layer switcher
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'Satellite';
  oLayer.Provider := mtpMapTilerSatellite;
  sgcMapApplyProvider(oLayer);
  oLayer.ApiKey := 'YOUR_MAPTILER_KEY';
  oLayer.Visible := False;

  // overlay: any tile server through a URL template
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'Hiking trails';
  oLayer.TileSourceType := mtsHTTP;
  oLayer.URLTemplate := 'https://tile.waymarkedtrails.org/hiking/{z}/{x}/{y}.png';
  oLayer.Attribution := '(c) waymarkedtrails.org';
  oLayer.IsBaseLayer := False;
  oLayer.Opacity := 0.8;

  sgcVCLMap1.Options.ShowLayerSwitcher := True;
end;
```

Caché de teselas

Establezca **CacheFolder** para conservar en disco todas las teselas descargadas, en un árbol z/x/y. Las teselas que ya están en la caché se dibujan sin ninguna solicitud al servidor, así que un mapa que el usuario ha visto una vez sigue funcionando sin conexión. El mapa también conserva en memoria las teselas recientes; **RefreshTiles** recarga las teselas de la vista. Una tesela que no se puede cargar dispara **OnTileError** con la capa, la dirección de la tesela y el texto del error, y **LastRenderMissingTiles** cuenta las teselas que faltaban en el último repintado.

Disponibilidad

Framework	Versión mínima
VCL, teselas de carpeta	Delphi 7
VCL, teselas de un servidor (proveedores, plantillas de URL, caché)	Delphi 2009
VCL y FireMonkey, MBTiles	Delphi XE7 (FireDAC SQLite)
FireMonkey	Delphi XE6
Lazarus (LCL)	Solo teselas de carpeta

Desde Delphi 2009 hasta XE7 las teselas se descargan con Indy 10, y https necesita las bibliotecas OpenSSL 1.0.2 (libeay32.dll y ssleay32.dll) junto al ejecutable. A partir de Delphi XE8 el mapa usa THTTPClient con la compatibilidad TLS del sistema operativo, y no necesita ninguna DLL. En Delphi 7, Delphi 2007 y Lazarus no hay descarga de teselas, así que use una carpeta de teselas.

Véase también

- TsgcVCLMap
- Teselas vectoriales
- Teselas sin conexión y el descargador de teselas

Teselas vectoriales

Una **tesela vectorial** almacena carreteras, edificios, agua y nombres de lugares como geometría en lugar de píxeles, en el formato Mapbox Vector Tile (MVT). sgcMaps decodifica las teselas y las dibuja por sí mismo, siguiendo un **estilo**: un JSON de estilo de MapLibre o de Mapbox GL que indica qué elementos dibujar, con qué colores y anchuras, y en qué niveles de zoom. Un mismo conjunto de teselas vectoriales puede así verse claro, oscuro o minimalista según el estilo, y las teselas se mantienen nítidas con cualquier densidad de píxeles.

Una capa vectorial

Añada una capa al mapa con uno de los tipos de origen vectorial:

- **mtsVectorHTTP**: teselas de un servidor. URLTemplate usa los mismos marcadores de posición que las teselas raster, por ejemplo https://api.maptiler.com/tiles/v3/{z}/{x}/{y}.pbf?key={apikey}, y ApiKey rellena {apikey}.
- **mtsVectorMBTiles**: teselas de un archivo MBTiles vectorial local establecido en TileSourcePath, como un archivo creado con tilemaker o planetiler.

VectorStyleFile es la ruta de un archivo JSON de estilo guardado como UTF-8. Cuando está vacío, el mapa usa el estilo integrado, un estilo claro para el esquema **OpenMapTiles** (agua, cobertura del suelo, uso del suelo, parques, edificios, carreteras, límites, cursos de agua y los nombres de lugares, carreteras y aguas). Ese esquema es el que usan MapTiler, OpenFreeMap y la mayoría de las teselas vectoriales de OpenStreetMap. Con **Retina** establecido en True, las teselas se renderizan a 512 píxeles en lugar de 256, para pantallas de alta densidad de píxeles. Las etiquetas de las capas de símbolos del estilo se recopilan por tesela y el mapa las dibuja en posición vertical, también cuando el mapa está rotado.

```
procedure TForm2.SetupVectorLayers;
var
  oLayer: TsgcMapLayer;
begin
  // vector tiles from a server, drawn with a MapLibre style file
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'MapTiler vector';
  oLayer.TileSourceType := mtsVectorHTTP;
  oLayer.URLTemplate :=
    'https://api.maptiler.com/tiles/v3/{z}/{x}/{y}.pbf?key={apikey}';
  oLayer.ApiKey := 'YOUR_MAPTILER_KEY';
  oLayer.Attribution := '(c) MapTiler (c) OpenStreetMap contributors';
  oLayer.VectorStyleFile := 'styles\dark.json';
  // renders every tile at 512 pixels
  oLayer.Retina := True;

  // the same data offline, from a vector MBTiles file and the built-in style
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'Offline vector';
  oLayer.TileSourceType := mtsVectorMBTiles;
  oLayer.TileSourcePath := 'C:\Maps\andorra.mbtiles';
  oLayer.VectorStyleFile := '';
  oLayer.Visible := False;
end;
```

El origen de teselas vectoriales

La capa crea un **TsgcMapTileSource_MVT** (unidad sgcMaps_TileSource_MVT). También puede crear uno usted mismo para renderizar teselas fuera de un mapa:

Miembro	Descripción
SourceKind	mvsHTTP o mvsMBTiles.

URLTemplate, ApiKey, UserAgent	El servidor de mvsHTTP.
Path	El archivo .mbtiles de mvsMBTiles.
StyleJSON, StyleFile	El estilo. StyleJSON prevalece sobre StyleFile, y con ambos vacíos Open carga el estilo integrado.
TileSize	256 (predeterminado) o 512 píxeles.
MaxDataZoom	El zoom más alto almacenado en los datos, 14 de forma predeterminada. Los niveles de zoom más profundos se dibujan a partir de las teselas de este nivel.
CacheSize	El número de teselas renderizadas que se conservan en memoria, 256 de forma predeterminada.
Open, Close	Open crea el lector y carga el estilo; genera una excepción si el estilo no es válido.
GetTile	Devuelve la tesela renderizada como imagen PNG.
GetTileLabels	Devuelve las etiquetas de una tesela, con su posición.
ClearCache	Descarta las teselas renderizadas. Llámelo después de cambiar el estilo o TileSize.
Style	El TsgcMapVectorStyle en uso, válido después de Open.

Decodificación y estilos

La unidad **sgcMaps_MVT** contiene los bloques de construcción. **TsgcMapVectorTile** decodifica una tesela a partir de bytes, de un flujo o de un archivo, sin comprimir o comprimida con gzip o zlib, en capas (**TsgcMapVectorLayer**) y elementos (**TsgcMapVectorFeature**) con su tipo de geometría, sus etiquetas y su geometría. **TsgcMapVectorStyle** carga un estilo con LoadFromJSON o LoadFromFile, o el integrado con LoadDefault.

```
procedure TForm2.ListRoadClasses(const aFileName: string);
var
  oTile: TsgcMapVectorTile;
  oLayer: TsgcMapVectorLayer;
  i: Integer;
begin
  oTile := TsgcMapVectorTile.Create;
  try
    // raw, gzip or zlib compressed protobuf
    oTile.LoadFromFile(aFileName);
    oLayer := oTile.LayerByName('transportation');
    if oLayer <> nil then
      for i := 0 to oLayer.FeatureCount - 1 do
        if oLayer.Features[i].GeomType = mvgLineString then
          memoLog.Lines.Add(oLayer.Features[i].GetTagAsString('class'));
  finally
    oTile.Free;
  end;
end;
```

La compatibilidad con estilos abarca los tipos de capa background, fill, line y symbol; las capas raster, fill-extrusion, circle, heatmap y hillshade se ignoran. Se evalúan los filtros, las funciones de zoom y las expresiones habituales (get, has, match, case, step, interpolate, coalesce y los operadores de comparación). No se usan iconos, sprites, patrones, halos de texto ni glifos.

Disponibilidad

Framework	Versión mínima
VCL y FireMonkey, teselas vectoriales desde un servidor	Delphi XE2 (FireMonkey desde XE6)
VCL y FireMonkey, MBTiles vectorial	Delphi XE7 (FireDAC SQLite)
Lazarus (LCL)	No disponible

Véase también

- Teselas, capas y proveedores
- TsgcVCLMap
- TsgcFMXMap

Marcadores y agrupación

Un **marcador** señala un punto del mapa. La colección **Markers** de TsgcVCLMap y de TsgcFMXMap contiene elementos **TsgcMapMarker**, y **AddMarker** añade uno con un texto de ventana emergente y una información sobre herramientas en una sola llamada.

Propiedades del marcador

Propiedad	Descripción
Latitude, Longitude	La posición del marcador.
PopupText	El texto de la ventana emergente que abre ShowPopup, normalmente desde OnMarkerClick. HidePopup la cierra y ActivePopupMarker es el marcador cuya ventana emergente está abierta.
TooltipText	El texto que se muestra mientras el ratón está sobre el marcador.
Color	El color del pin predeterminado. MarkerOptions.DefaultColor, DefaultWidth y DefaultHeight definen el pin de todos los marcadores.
IconBitmap	Un icono personalizado, cargado con LoadFromFile o LoadFromStream desde una imagen PNG (Delphi 2009 y posteriores, y Lazarus), JPEG o BMP. Se almacena en el formulario, de modo que un icono establecido en tiempo de diseño no necesita ningún archivo en tiempo de ejecución.
AnchorX, AnchorY	El punto del icono que toca la coordenada, como fracción de su ancho y de su alto. El valor predeterminado, 0,5 y 1, es el centro inferior.
Draggable	Permite al usuario mover el marcador. OnMarkerDragStart, OnMarkerDrag y OnMarkerDragEnd siguen el arrastre.
Visible, Tag, Data	Visibilidad, y un entero y una cadena para sus propios datos.

MarkerAtScreen devuelve el marcador situado en un punto del control, **FitToMarkers** ajusta el zoom y centra la vista en todos los marcadores, y **ClearMarkers** los elimina todos.

```
procedure TForm2.AddStore;
var
  oMarker: TsgcMapMarker;
begin
  oMarker := sgcVCLMap1.Markers.Add;
  oMarker.Latitude := 41.3874;
  oMarker.Longitude := 2.1686;
  oMarker.PopupText := 'Barcelona store';
  oMarker.TooltipText := 'Open from 9:00 to 20:00';
  oMarker.Data := 'store-012';
  // a custom icon whose bottom center touches the coordinate
  oMarker.IconBitmap.LoadFromFile('icons\store.png');
  oMarker.AnchorX := 0.5;
  oMarker.AnchorY := 1.0;
  oMarker.Draggable := True;
  sgcVCLMap1.OnMarkerDragEnd := sgcVCLMap1MarkerDragEnd;
end;

procedure TForm2.sgcVCLMap1MarkerDragEnd(Sender: TObject;
```

```
  aMarker: TsgcMapMarker);
begin
  memoLog.Lines.Add(Format('%s moved to %.5f, %.5f',
    [aMarker.Data, aMarker.Latitude, aMarker.Longitude]));
end;
```

Agrupación

Miles de marcadores hacen que un mapa sea lento e ilegible cuando se aleja el zoom. Con **MarkerOptions.Clustering.Enabled** establecido en True, los marcadores cercanos se fusionan en un círculo que muestra cuántos marcadores contiene. La pantalla se divide en celdas cuadradas de **GridSize** píxeles, y una celda con al menos **MinClusterSize** marcadores se convierte en una agrupación dibujada en el centro de sus marcadores. Por encima de **MaxZoom** (18 de forma predeterminada) los marcadores nunca se agrupan. **Color** y **FontColor** definen el aspecto de los círculos.
Un clic en una agrupación dispara **OnClusterClick** con el número de marcadores y la posición de la agrupación. Cuando el manejador deja aHandled en False y **ZoomOnClick** es True, el valor predeterminado, el mapa acerca entonces el zoom hasta que los marcadores de la agrupación se separan. **ClusterAtScreen** encuentra la agrupación situada en un punto del control, y **DrawnClusterCount** y **DrawnMarkerCount** informan de lo que dibujó el último repintado.

```
procedure TForm2.AddManyMarkers;
var
  i: Integer;
begin
  sgcVCLMap1.MarkerOptions.Clustering.Enabled := True;
  sgcVCLMap1.MarkerOptions.Clustering.GridSize := 60;
  // from zoom 17 on, every marker is drawn on its own
  sgcVCLMap1.MarkerOptions.Clustering.MaxZoom := 16;
  sgcVCLMap1.OnClusterClick := sgcVCLMap1ClusterClick;

  sgcVCLMap1.Markers.BeginUpdate;
  try
    for i := 1 to 2000 do
      sgcVCLMap1.AddMarker(40 + Random * 3, -4 + Random * 5,
        'Point ' + IntToStr(i));
  finally
    sgcVCLMap1.Markers.EndUpdate;
  end;
  sgcVCLMap1.FitToMarkers;
end;

procedure TForm2.sgcVCLMap1ClusterClick(Sender: TObject; aCount: Integer;
  aLatitude, aLongitude: Double; var aHandled: Boolean);
begin
  memoLog.Lines.Add(Format('%d markers near %.4f, %.4f',
    [aCount, aLatitude, aLongitude]));
end;
```

Envuelva las adiciones masivas en Markers.BeginUpdate y Markers.EndUpdate, para que el mapa se repinte una vez en lugar de una vez por marcador. La propia función de agrupación, **sgcMapClusterPoints**, es pública en sgcMaps_Geometry para sus propios puntos de pantalla.

Disponibilidad

Framework	Versión mínima
VCL	Delphi 7
FireMonkey	Delphi XE6
Lazarus (LCL)	Sí

Véase también

- TsgcVCLMap
- Formas y etiquetas
- Mapas de calor

Formas y etiquetas

Además de los marcadores, TsgcVCLMap y TsgcFMXMap dibujan polilíneas, polígonos, rectángulos, círculos y etiquetas de texto sobre las teselas. Cada tipo tiene su propia colección y un método Add en el control que crea un elemento y lo devuelve.

Colección	Método Add	Elemento	Propiedades principales
Polylines	AddPolyline	TsgcMapPolyline	AddPoint, ClearPoints, PointCount, Color, Width, Opacity, Dashed, Geodesic, Editable
Polygons	AddPolygon	TsgcMapPolygon	AddPoint, Holes, FillColor, FillOpacity, StrokeColor, StrokeWidth, Editable
Rectangles	AddRectangle	TsgcMapRectangle	NorthLatitude, WestLongitude, SouthLatitude, EastLongitude, FillColor, FillOpacity, StrokeColor, StrokeWidth, Editable
Circles	AddCircle	TsgcMapCircle	CenterLat, CenterLon, RadiusMeters, FillColor, FillOpacity, StrokeColor, StrokeWidth
Labels	AddLabel	TsgcMapLabel	Latitude, Longitude, Text, FontSize, FontColor, BackgroundColor, ShowBackground

Todos los elementos tienen además Visible, y las formas tienen un Tag para su propio uso.

Polilíneas

Una polilínea une sus puntos con líneas rectas en el mapa. Con **Dashed** se dibuja discontinua, y con **Geodesic** cada segmento sigue el círculo máximo, el camino más corto sobre el globo, que es la opción adecuada para vuelos y grandes distancias. **FitToPolyline** ajusta la vista a una polilínea, y las rutas de los componentes de indicaciones y de enrutamiento también se añaden como polilíneas.

Polígonos con huecos

Un polígono se rellena dentro de su anillo exterior. **Holes** añade anillos interiores que se dejan vacíos: **AddHole** devuelve el índice de un hueco nuevo, y **AddPoint** le añade un vértice. Los polígonos con huecos de GeoJSON y KML se importan del mismo modo.

```
procedure TForm2.AddShapes;
var
  oLine: TsgcMapPolyline;
  oPolygon: TsgcMapPolygon;
  iHole: Integer;
begin
  // a dashed great circle from Madrid to New York
  oLine := sgcVCLMap1.AddPolyline;
  oLine.AddPoint(40.4722, -3.5608);
  oLine.AddPoint(40.6413, -73.7781);
  oLine.Geodesic := True;
  oLine.Dashed := True;
  oLine.Color := clNavy;
  oLine.Width := 3;

  // a polygon with a hole, editable with the mouse
  oPolygon := sgcVCLMap1.AddPolygon;
  oPolygon.AddPoint(40.425, -3.700);
  oPolygon.AddPoint(40.425, -3.675);
```

```
  oPolygon.AddPoint(40.405, -3.675);
  oPolygon.AddPoint(40.405, -3.700);
  iHole := oPolygon.Holes.AddHole;
  oPolygon.Holes.AddPoint(iHole, 40.418, -3.692);
  oPolygon.Holes.AddPoint(iHole, 40.418, -3.683);
  oPolygon.Holes.AddPoint(iHole, 40.412, -3.683);
  oPolygon.Holes.AddPoint(iHole, 40.412, -3.692);
  oPolygon.FillColor := clGreen;
  oPolygon.FillOpacity := 0.3;
  oPolygon.Editable := True;

  sgcVCLMap1.AddRectangle(40.43, -3.72, 40.42, -3.70).FillOpacity := 0.2;
  sgcVCLMap1.AddCircle(40.4168, -3.7038, 250).StrokeColor := clRed;
  sgcVCLMap1.AddLabel(40.4168, -3.7038, 'Puerta del Sol').FontSize := 12;
  sgcVCLMap1.OnShapeEdited := sgcVCLMap1ShapeEdited;
end;

procedure TForm2.sgcVCLMap1ShapeEdited(Sender: TObject; aItem: TCollectionItem);
begin
  if aItem is TsgcMapPolygon then
    memoLog.Lines.Add(Format('Polygon %d has %d points now',
      [aItem.Index, TsgcMapPolygon(aItem).PointCount]));
end;
```

Edición

Establezca **Editable** en True en una polilínea, un polígono o un rectángulo para que el usuario pueda remodelarlo con el ratón: un vértice se mueve arrastrándolo, el tirador situado en medio de un segmento inserta un vértice nuevo, y un clic derecho en un vértice lo elimina. Cuando termina una edición, el mapa dispara **OnShapeEdited** con el elemento editado. Los círculos y las etiquetas no son editables.

Disponibilidad

Framework	Versión mínima
VCL	Delphi 7
FireMonkey	Delphi XE6
Lazarus (LCL)	Sí

Véase también

- TsgcVCLMap
- Marcadores y agrupación
- Funciones de geometría
- GeoJSON, GPX y KML

Mapas de calor

Un **mapa de calor** muestra dónde se concentran puntos ponderados: lecturas de sensores, ventas, incidentes o posiciones GPS. Cada punto reparte su peso sobre un círculo, los pesos se suman y el total se colorea de frío a caliente.
La colección **Heatmaps** de TsgcVCLMap y de TsgcFMXMap contiene elementos **TsgcMapHeatmap**. Añada los puntos con **AddPoint**(latitud, longitud, peso); el peso es 1 cuando se omite. Count, Points y Clear dan acceso a los puntos, y un registro **TsgcMapHeatmapPoint** contiene uno de ellos.

Propiedad	Predeterminado	Descripción
Radius	25	El radio de influencia de cada punto, en píxeles.
Opacity	0.6	La opacidad del color más caliente.
MaxIntensity	0	La intensidad que se dibuja con el color más caliente. 0 escala los colores a la intensidad más alta dibujada; un valor fijo mantiene los colores comparables entre vistas.
Gradient	mhgDefault	mhgDefault (azul, cian, lima, amarillo, rojo), mhgFire, mhgCool o mhgGrayscale.
Visible	True	Muestra u oculta el mapa de calor.

```
procedure TForm2.ShowReadings;
var
  oHeatmap: TsgcMapHeatmap;
  i: Integer;
begin
  // FReadings: array of TsgcMapHeatmapPoint, loaded elsewhere
  oHeatmap := sgcVCLMap1.Heatmaps.Add;
  oHeatmap.Radius := 30;
  oHeatmap.Gradient := mhgFire;
  oHeatmap.Opacity := 0.7;
  // 0 scales the colors to the hottest spot drawn
  oHeatmap.MaxIntensity := 0;
  for i := 0 to High(FReadings) do
    oHeatmap.AddPoint(FReadings[i].Latitude, FReadings[i].Longitude,
      FReadings[i].Weight);
end;
```

El mapa de calor se dibuja en píxeles de pantalla, por lo que su forma sigue el zoom: al acercar, los puntos calientes se separan. La función de renderizado, **sgcMapRenderHeatmap** en sgcMaps_Geometry, también es pública y rellena un búfer ARGB a partir de sus propios puntos de pantalla.

Disponibilidad

Framework	Versión mínima
VCL	Delphi 7
FireMonkey	Delphi XE6
Lazarus (LCL)	Sí

Véase también

- TsgcVCLMap
- Marcadores y agrupación

GeoJSON, GPX y KML

sgcMaps lee y escribe los tres formatos habituales de intercambio de datos geográficos. Las funciones de importación añaden elementos a las colecciones de un mapa, y las funciones de exportación escriben los elementos de un mapa, de modo que se puede pasar cualquier colección de TsgcVCLMap o de TsgcFMXMap.

Formato	Unidad	Importación	Exportación
GeoJSON (RFC 7946)	sgcMaps_GeoJSON	Point y MultiPoint a marcadores, LineString y MultiLineString a polilíneas, Polygon y MultiPolygon a polígonos con huecos, GeometryCollection, Feature y FeatureCollection.	Un FeatureCollection con propiedades simplestyle. Los rectángulos se escriben como polígonos.
GPX	sgcMaps_GPX	Puntos de paso a marcadores, recorridos y rutas a polilíneas.	Marcadores a puntos de paso y polilíneas a recorridos. Los polígonos y los rectángulos se escriben como recorridos cerrados cuando ExportPolygonsAsTracks es True.
KML	sgcMaps_KML	Point a marcadores, LineString a polilíneas, Polygon con sus límites interiores a polígonos, con los colores de los estilos en línea y compartidos.	Un Placemark por elemento con un estilo en línea. Los rectángulos se escriben como polígonos.

Cada formato tiene una clase (**TsgcMapGeoJSON**, **TsgcMapGPX**, **TsgcMapKML**) con LoadFromString, LoadFromFile, AsText, SaveToStream y SaveToFile, y las mismas operaciones como funciones sueltas: **sgcMapLoadGeoJSON**, **sgcMapLoadGeoJSONFile**, **sgcMapGeoJSONAsText**, **sgcMapSaveGeoJSONFile** y las funciones equivalentes de GPX y KML. Una colección que se pasa como nil se omite.

```
procedure TForm2.ImportAndExport;
begin
  // GeoJSON: points, lines and polygons with holes (Delphi XE6 and later)
  sgcMapLoadGeoJSONFile('parks.geojson', sgcVCLMap1.Markers,
    sgcVCLMap1.Polylines, sgcVCLMap1.Polygons);
  // GPX: waypoints become markers, tracks and routes become polylines
  sgcMapLoadGPXFile('ride.gpx', sgcVCLMap1.Markers, sgcVCLMap1.Polylines);
  // KML: placemarks with their colors
  sgcMapLoadKMLFile('areas.kml', sgcVCLMap1.Markers, sgcVCLMap1.Polylines,
    sgcVCLMap1.Polygons);
  sgcVCLMap1.FitToMarkers;

  // everything the map shows, rectangles included
  sgcMapSaveGeoJSONFile('export.geojson', sgcVCLMap1.Markers,
    sgcVCLMap1.Polylines, sgcVCLMap1.Polygons, sgcVCLMap1.Rectangles);
  sgcMapSaveKMLFile('export.kml', sgcVCLMap1.Markers, sgcVCLMap1.Polylines,
    sgcVCLMap1.Polygons, sgcVCLMap1.Rectangles);
end;
```

Estadísticas de recorridos GPX

TsgcMapGPX guarda estadísticas de cada recorrido del último archivo que cargó. **TrackStatsCount** y **TrackStats** devuelven un registro **TsgcMapGPXTrackStats** con el nombre, el número de puntos, la distancia en metros, la elevación mínima y máxima, y el ascenso y el descenso totales. Con **ColorByElevation** establecido en True, cada recorrido se divide en tramos coloreados según la elevación, desde LowColor pasando por MidColor hasta HighColor de **ElevationGradient** (un TsgcMapElevationGradient).

```
procedure TForm2.ShowTrackStats;
var
  oGPX: TsgcMapGPX;
  oStats: TsgcMapGPXTrackStats;
  i: Integer;
begin
  oGPX := TsgcMapGPX.Create;
  try
    // splits every track into pieces colored by elevation
    oGPX.ColorByElevation := True;
    oGPX.LoadFromFile('ride.gpx', sgcVCLMap1.Markers, sgcVCLMap1.Polylines);
    for i := 0 to oGPX.TrackStatsCount - 1 do
    begin
      oStats := oGPX.TrackStats[i];
      memoLog.Lines.Add(Format('%s: %.1f km, up %.0f m, down %.0f m',
        [oStats.Name, oStats.Distance / 1000, oStats.TotalAscent,
        oStats.TotalDescent]));
    end;
  finally
    oGPX.Free;
  end;
end;
```

Disponibilidad

Framework	Versión mínima
GPX y KML, VCL y FireMonkey	Delphi 7 (FireMonkey desde XE6)
GeoJSON, VCL y FireMonkey	Delphi XE6 (System.JSON)
Lazarus (LCL)	No disponible

GPX y KML se analizan con TXMLDocument: MSXML en Windows, y OmniXML en las demás plataformas a partir de Delphi XE7.

Véase también

- Formas y etiquetas
- Marcadores y agrupación
- TsgcVCLMap

Funciones de geometría

La unidad **sgcMaps_Geometry** contiene funciones que trabajan con coordenadas, sin ningún control de mapa de por medio, por lo que sirven tanto a un servicio o a una aplicación de consola como a un formulario. Las coordenadas son grados WGS84, las distancias son metros sobre una esfera con el radio medio de la Tierra (SGC_MAP_EARTH_RADIUS, 6371008,8 m) y los rumbos son grados en sentido horario desde el norte verdadero, de 0 a 360.

Función	Devuelve
sgcMapDistance	La distancia de círculo máximo entre dos puntos, en metros.
sgcMapInitialBearing	El rumbo inicial de un punto a otro.
sgcMapDestinationPoint	El punto que se alcanza desde un punto de partida con un rumbo y una distancia.
sgcMapGeodesicInterpolate	Los puntos del círculo máximo entre dos puntos, divididos en un número de segmentos.
sgcMapPointInPolygon	True cuando un punto está dentro de un anillo.
sgcMapPointInPolygonWithHoles	True cuando un punto está dentro de un anillo exterior y fuera de todos sus huecos.
sgcMapSimplify	Una línea con menos puntos (Douglas-Peucker), dentro de una tolerancia en metros.
sgcMapNearestPointOnPolyline	La distancia de un punto a una línea, con el punto más cercano y el índice de su segmento.

Las matrices de puntos usan **TsgcMapCoordArray**, una matriz del registro **TsgcMapCoord**, y **sgcMapCoord** construye una coordenada.

```
procedure TForm2.GeometryExamples;
var
  dDistance, dBearing, dLat, dLon: Double;
  vArea, vTrack: TsgcMapCoordArray;
  iSegment: Integer;
begin
  // Madrid to Barcelona: meters and initial bearing
  dDistance := sgcMapDistance(40.4168, -3.7038, 41.3874, 2.1686);
  dBearing := sgcMapInitialBearing(40.4168, -3.7038, 41.3874, 2.1686);
  memoLog.Lines.Add(Format('%.1f km, bearing %.1f', [dDistance / 1000,
    dBearing]));

  // the point 10 km to the north east of Madrid, inside the area?
  sgcMapDestinationPoint(40.4168, -3.7038, 45, 10000, dLat, dLon);
  SetLength(vArea, 3);
  vArea[0] := sgcMapCoord(40.3, -3.9);
  vArea[1] := sgcMapCoord(40.7, -3.5);
  vArea[2] := sgcMapCoord(40.3, -3.3);
  if sgcMapPointInPolygon(dLat, dLon, vArea) then
    memoLog.Lines.Add('Inside the area');

  // a great circle track, simplified to 50 m, and the distance to it
  vTrack := sgcMapGeodesicInterpolate(40.4168, -3.7038, 41.3874, 2.1686, 64);
  vTrack := sgcMapSimplify(vTrack, 50);
  dDistance := sgcMapNearestPointOnPolyline(sgcVCLMap1.CenterLatitude,
    sgcVCLMap1.CenterLongitude, vTrack, iSegment, dLat, dLon);
```

```
  memoLog.Lines.Add(Format('%.0f m to segment %d', [dDistance, iSegment]));
end;
```

Límites y matemática de teselas

FitBounds de los controles de mapa recibe un cuadro (norte, oeste, sur, este) y lo muestra con el mayor zoom que cabe, conservando un margen en píxeles; una longitud oeste mayor que la este cruza el antimeridiano. FitToMarkers y FitToPolyline hacen lo mismo con los marcadores y con una polilínea. La unidad sgcMaps_Classes añade la matemática de teselas de la proyección Web Mercator: **sgcMapLatLonToTileF** y **sgcMapTileFToLatLon** convierten entre coordenadas y números de tesela fraccionarios, **sgcMapLatLonToPixel** y **sgcMapPixelToLatLon** entre coordenadas y píxeles del mundo, y **sgcMapTileCount** da el número de teselas por lado en un zoom.

```
procedure TForm2.ShowSpain;
var
  dTileX, dTileY: Double;
begin
  // the largest zoom that shows the whole box, with 20 pixels of margin
  sgcVCLMap1.FitBounds(43.8, -9.3, 36.0, 3.3, 20);

  // the tile under the center of the view
  if sgcMapLatLonToTileF(sgcVCLMap1.CenterLatitude,
    sgcVCLMap1.CenterLongitude, sgcVCLMap1.Zoom, dTileX, dTileY) then
    memoLog.Lines.Add(Format('Tile %d/%d/%d', [sgcVCLMap1.Zoom, Trunc(dTileX),
      Trunc(dTileY)]));
end;
```

Disponibilidad

Framework	Versión mínima
VCL, FireMonkey, consola y servicios	Delphi 7
Lazarus	Sí

Véase también

- Formas y etiquetas
- Marcadores y agrupación
- TsgcVCLMap

Renderizado de mapas estáticos

Un control de mapa puede renderizar sin ventana. Un TsgcVCLMap creado sin padre se pinta en un mapa de bits cuando se le pide, lo que convierte una aplicación de consola, un servidor web o un servicio de Windows en un renderizador de mapas estáticos: miniaturas para un informe, un mapa adjunto a un correo electrónico o una imagen servida por HTTP.

Métodos de renderizado

Método	Descripción
RenderToBitmap(aBitmap, aInclude-Controls)	Pinta el mapa en un mapa de bits de su mismo tamaño. Un mapa de bits vacío toma el tamaño del control.
SaveToPNG(aFileName, aWidth, aHeight, aIncludeControls)	Escribe un archivo PNG del tamaño indicado.
SaveToBMP(aFileName, aWidth, aHeight, aIncludeControls)	Escribe un archivo BMP del tamaño indicado.

La imagen muestra las teselas, las superposiciones, la barra de escala y la atribución. Los botones de zoom, el selector de capas y la brújula se omiten salvo que aIncludeControls sea True. Las teselas se leen durante el renderizado, y las teselas de un servidor se descargan en ese momento, así que la llamada se bloquea hasta que llegan. Después, **LastRenderMissingTiles** indica cuántas teselas no se pudieron cargar, y **AttributionText** devuelve el texto de copyright que se debe mostrar junto a la imagen.

```
program StaticMap;

{$APPTYPE CONSOLE}

uses
  SysUtils,
  sgcMaps_Classes, sgcMaps_Providers, sgcMaps_VCL;

var
  oMap: TsgcVCLMap;
  oLayer: TsgcMapLayer;
begin
  // a map control with no parent: no window, it only renders
  oMap := TsgcVCLMap.Create(nil);
  try
    oMap.Options.ShowZoomControls := False;
    oLayer := oMap.Layers.Add;
    oLayer.Provider := mtpOpenStreetMap;
    sgcMapApplyProvider(oLayer);
    oLayer.CacheFolder := ExtractFilePath(ParamStr(0)) + 'tile_cache';

    oMap.AddMarker(48.8584, 2.2945, 'Eiffel Tower');
    oMap.SetCenter(48.8584, 2.2945);
    oMap.Zoom := 15;
    // the tiles are downloaded while the image is rendered
    oMap.SaveToPNG('eiffel.png', 1024, 768);
    Writeln('Missing tiles: ', oMap.LastRenderMissingTiles);
    Writeln(oMap.AttributionText);
  finally
    oMap.Free;
  end;
end.
```

En un formulario, los mismos métodos exportan la vista que está mirando el usuario:

```
procedure TForm2.PreviewMap;
var
  oBitmap: TBitmap;
begin
  oBitmap := TBitmap.Create;
  try
    // the bitmap size is the size of the image; the controls are left out
    oBitmap.Width := 1200;
    oBitmap.Height := 900;
    sgcVCLMap1.RenderToBitmap(oBitmap);
    imgPreview.Picture.Assign(oBitmap);
  finally
    oBitmap.Free;
  end;
end;
```

El rasterizador

Las teselas vectoriales se dibujan con **TsgcMapRasterizer** (unidad sgcMaps_Raster), un rasterizador por software con suavizado que no depende de VCL ni de FireMonkey, y que puede usar para dibujar sus propias imágenes. Rellena y traza trayectos (BeginPath, MoveTo, LineTo, ClosePath, FillPath con la regla de no cero o de par-impar, StrokePath con uniones, remates y guiones), recorta a un rectángulo y guarda el resultado con SaveToBMPFile y SaveToPNGFile. Los colores son valores $AARRGGBB directos.

```
procedure TForm2.DrawWarningIcon;
var
  oRaster: TsgcMapRasterizer;
begin
  oRaster := TsgcMapRasterizer.Create(64, 64);
  try
    oRaster.Clear($00000000);
    oRaster.BeginPath;
    oRaster.MoveTo(32, 4);
    oRaster.LineTo(60, 60);
    oRaster.LineTo(4, 60);
    oRaster.ClosePath;
    // colors are $AARRGGBB
    oRaster.FillPath($FFFFC107);
    oRaster.StrokePath($FF000000, 3, sjRound);
    oRaster.SaveToPNGFile('warning.png');
  finally
    oRaster.Free;
  end;
end;
```

Disponibilidad

Framework	Versión mínima
VCL, salida BMP	Delphi 7
VCL, salida PNG	Delphi 2009
FireMonkey	Delphi XE6
Lazarus (LCL)	Sí, PNG y BMP

Las teselas de un servidor necesitan Delphi 2009 o posterior y no están disponibles en Lazarus, donde el control dibuja únicamente teselas de carpeta; véase Teselas, capas y proveedores. El rasterizador comprime los archivos PNG con System.ZLib desde Delphi XE2, y los escribe sin comprimir en versiones anteriores.

Véase también

- TsgcVCLMap
- TsgcFMXMap
- Teselas, capas y proveedores

Teselas sin conexión y el descargador de teselas

Un mapa que debe funcionar sin conexión de red, en un vehículo, una planta industrial o un barco, dibuja sus teselas a partir de archivos locales. sgcMaps lee dos tipos de paquete de teselas sin conexión:

- Una **carpeta** (TileSourceType mtsFolder): TileSourcePath contiene una subcarpeta por zoom, una subcarpeta por columna y un archivo por tesela, con el nombre z\x\y.png, .jpg o .jpeg. Es la estructura que escriben TsgcMapTileDownloader, la caché en disco de una capa y muchas herramientas de teselas.
- Un archivo **MBTiles** (TileSourceType mtsMBTiles): un único archivo SQLite que contiene todas las teselas. Es fácil de copiar y lo producen herramientas como MapTiler, TileMill y gdal. Los archivos MBTiles vectoriales se describen en Teselas vectoriales.

```
procedure TForm2.UseOfflineTiles;
var
  oLayer: TsgcMapLayer;
begin
  sgcVCLMap1.Layers.Clear;
  // a z/x/y folder, for example the output of TsgcMapTileDownloader
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'Tile folder';
  oLayer.TileSourceType := mtsFolder;
  oLayer.TileSourcePath := 'C:\Maps\andorra';
  oLayer.MaxZoom := 15;

  // a raster MBTiles file (Delphi XE7 and later)
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'MBTiles';
  oLayer.TileSourceType := mtsMBTiles;
  oLayer.TileSourcePath := 'C:\Maps\andorra.mbtiles';
  oLayer.Visible := False;

  sgcVCLMap1.Options.ShowLayerSwitcher := True;
end;
```

Descarga de un paquete de teselas

TsgcMapTileDownloader (unidad sgcMaps_TileDownloader) llena una carpeta con todas las teselas de un área y de un rango de niveles de zoom.

Miembro	Descripción
URLTemplate	El servidor de teselas, con los marcadores de posición {z} {x} {y}. A propósito no tiene valor predeterminado.
OutputPath	La carpeta raíz del árbol z/x/y.
UserAgent	La cabecera User-Agent que se envía con cada solicitud.
DelayMS	La pausa entre dos descargas, 250 ms de forma predeterminada.
Timeout	El tiempo de espera de cada solicitud, 15000 ms de forma predeterminada.
CountTiles	El número de teselas de un área y de un rango de zoom, para comprobar el tamaño antes de descargar.
GetTileRange	La primera y la última columna y fila de teselas de un área en un zoom.

Download	Descarga las teselas. Las teselas que ya están en disco se omiten, así que se puede reanudar una descarga interrumpida.
OnProgress	Informa del total de teselas, de las descargadas, de las omitidas y de las fallidas, y del zoom actual. Establezca aCancel en True para detener el proceso.

Download es síncrono: bloquea el hilo que lo llama y dispara OnProgress en ese mismo hilo. Llámelo desde un hilo de trabajo para mantener la interfaz de usuario receptiva. Genera una excepción cuando URLTemplate o OutputPath están vacíos.

```
procedure TForm2.DownloadAndorra;
var
  oDownloader: TsgcMapTileDownloader;
begin
  oDownloader := TsgcMapTileDownloader.Create;
  try
    // your own tile server, or a provider whose terms allow bulk downloads
    oDownloader.URLTemplate := 'https://tiles.example.com/{z}/{x}/{y}.png';
    oDownloader.OutputPath := 'C:\Maps\andorra';
    oDownloader.UserAgent := 'MyMapApp/1.0 (contact@example.com)';
    oDownloader.DelayMS := 250;
    oDownloader.OnProgress := DownloadProgress;
    memoLog.Lines.Add(Format('%d tiles', [oDownloader.CountTiles(42.43, 1.41,
      42.66, 1.79, 10, 15)]));
    oDownloader.Download(42.43, 1.41, 42.66, 1.79, 10, 15);
  finally
    oDownloader.Free;
  end;
end;

procedure TForm2.DownloadProgress(Sender: TObject; aTotalTiles, aDownloaded,
  aSkipped, aFailed: Int64; aCurrentZoom: Integer; var aCancel: Boolean);
begin
  lblProgress.Caption := Format('Zoom %d: %d of %d tiles, %d failed',
    [aCurrentZoom, aDownloaded + aSkipped + aFailed, aTotalTiles, aFailed]);
  Application.ProcessMessages;
  aCancel := FCancelDownload;
end;
```

Tenga en cuenta las políticas de uso de teselas. La mayoría de los servidores públicos de teselas, incluido tile.openstreetmap.org, prohíben las descargas masivas y la precarga sin conexión. Dirija el descargador a su propio servidor de teselas o a un proveedor cuyas condiciones lo permitan, y mantenga DelayMS por encima de cero. El número de teselas se multiplica por cuatro con cada nivel de zoom, así que compruebe CountTiles antes de descargar niveles profundos.

Disponibilidad

Framework	Versión mínima
Teselas de carpeta	Delphi 7 y Lazarus
MBTiles	Delphi XE7 (FireDAC SQLite)
TsgcMapTileDownloader	Delphi 2009 (Indy 10; THTTPClient desde Delphi XE8)
Lazarus (LCL), MBTiles y el descargador	No disponible

Véase también

- Teselas, capas y proveedores
- Teselas vectoriales

- Instalación

Lector de OpenStreetMap

Los datos de OpenStreetMap se distribuyen como archivos **.osm.pbf** (Protocol Buffers, compactos y rápidos) y como archivos XML **.osm**. Geofabrik y otros espejos publican extractos por país o región. **TsgcMapOSMReader** (unidad sgcMaps_OSM) lee ambos formatos como un flujo: nunca construye el documento entero en memoria, así que lee archivos de cualquier tamaño y entrega cada elemento mediante un evento mientras se lee el archivo.

Miembro	Descripción
ReadFile, ReadStream	Lee un archivo o un flujo. El formato se detecta a partir del contenido; DetectFormat lo devuelve sin leer.
ReadNodes, ReadWays, ReadRelations	Qué elementos se entregan, todos True de forma predeterminada. Desactivar lo que no necesita hace la lectura más rápida.
SkipNodeTags	Omite las etiquetas de los nodos, para los lectores que solo necesitan coordenadas.
OnNode	Un nodo con su id, su latitud, su longitud y sus etiquetas.
OnWay	Una vía con su id, los ids de sus nodos y sus etiquetas.
OnRelation	Una relación con su id, sus miembros (TsgcMapOSMMember: tipo, referencia y rol) y sus etiquetas.
OnFilterWay	Se llama antes de entregar una vía. Establezca aAccept en False para omitirla.
OnProgress	Los bytes leídos y el tamaño total.

Las etiquetas llegan en un objeto **TsgcMapOSMTags** con Count, Keys, Values, Value, Has e IndexOf. El objeto de etiquetas y las matrices que se pasan a los eventos se reutilizan para el elemento siguiente, así que copie lo que necesite dentro del manejador.

```
procedure TForm2.ShowRestaurants(const aFileName: string);
var
  oReader: TsgcMapOSMReader;
begin
  FCount := 0;
  oReader := TsgcMapOSMReader.Create;
  try
    // only the nodes are needed
    oReader.ReadWays := False;
    oReader.ReadRelations := False;
    oReader.OnNode := OSMNode;
    // .osm.pbf or .osm XML, detected from the content
    oReader.ReadFile(aFileName);
  finally
    oReader.Free;
  end;
  memoLog.Lines.Add(Format('%d restaurants', [FCount]));
end;

procedure TForm2.OSMNode(Sender: TObject; aId: Int64; aLat, aLon: Double;
  aTags: TsgcMapOSMTags);
begin
  // aTags is reused for the next node: copy what you keep
  if aTags.Value('amenity') = 'restaurant' then
  begin
    Inc(FCount);
    sgcVCLMap1.AddMarker(aLat, aLon, aTags.Value('name'));
```

```
  end;
end;
```

Se admiten bloques PBF comprimidos con zlib o almacenados sin comprimir. Los bloques comprimidos con lzma, lz4, zstd o bzip2, y los archivos que requieren características distintas de OsmSchema-V0.6 y DenseNodes, generan **EsgcMapOSMError**. El lector es la entrada de TsgcMapRouteGraphBuilder, véase Grafos de rutas y enrutamiento sin conexión.

Disponibilidad

Framework	Versión mínima
VCL, FireMonkey, consola y servicios	Delphi XE2 (System.ZLib)
Lazarus	No disponible

Véase también

- Grafos de rutas y enrutamiento sin conexión
- TsgcMapOfflineRouter

Índice de geocodificación sin conexión

La geocodificación sin conexión de red necesita una base de datos local de nombres de lugares. sgcMaps crea una a partir de los volcados gratuitos de **GeoNames** y la consulta con SQLite, en dos pasos:

1. TsgcMapGeoIndexBuilder lee un volcado de GeoNames (cities500.zip, cities1000.zip, un archivo de país como ES.zip, allCountries.zip o el archivo .txt extraído), opcionalmente con admin1CodesASCII.txt y countryInfo.txt para los nombres de regiones y de países, y escribe una base de datos SQLite indexada. FeatureClasses selecciona los tipos de lugares que se conservan, P para los lugares poblados y A para las áreas administrativas.
2. TsgcMapOfflineGeocoder abre esa base de datos y responde a Geocode (un nombre de lugar, seguido opcionalmente de un país), ReverseGeocode y ReverseGeocodeCountry.

El índice se crea una sola vez, en una máquina de desarrollo o en el momento de la instalación, y se distribuye con la aplicación. Los resultados usan la misma clase **TsgcMapGeocodingResults** que el TsgcMapGeocoder en línea, así que el código que los muestra sirve para ambos.

Geocodificación inversa con un clic

ReverseGeocode devuelve los lugares situados a menos de 50 km de una coordenada, primero los lugares poblados y cada grupo ordenado por distancia, y ReverseGeocodeCountry devuelve el código ISO del país del lugar más cercano.

```
procedure TForm2.MapClickNearestPlace(Sender: TObject; aLat, aLon: Double);
var
  oResults: TsgcMapGeocodingResults;
begin
  // DatabaseFile was built by TsgcMapGeoIndexBuilder
  if not sgcMapOfflineGeocoder1.Active then
  begin
    sgcMapOfflineGeocoder1.DatabaseFile := 'geonames.db';
    sgcMapOfflineGeocoder1.Open;
  end;
  oResults := TsgcMapGeocodingResults.Create;
  try
    if sgcMapOfflineGeocoder1.ReverseGeocode(aLat, aLon, oResults) and
      (oResults.Count > 0) then
      memoLog.Lines.Add('Nearest place: ' + oResults[0].DisplayName)
    else
      memoLog.Lines.Add('No place within 50 km');
  finally
    oResults.Free;
  end;
end;
```

Cada componente mantiene una conexión SQLite, así que use un componente desde un solo hilo a la vez; cree uno por hilo para consultas en paralelo.

Disponibilidad

Framework	Versión mínima
VCL, FireMonkey, consola y servicios	Delphi XE7 (FireDAC SQLite)
Lazarus	No disponible

Véase también

- TsgcMapGeoIndexBuilder

- TsgcMapOfflineGeocoder
- TsgcMapGeocoder

Grafos de rutas y enrutamiento sin conexión

El cálculo de rutas sin conexión de red necesita la red viaria en disco, en una forma en la que un enrutador pueda buscar con rapidez. sgcMaps lo hace en dos pasos:

1. **TsgcMapRouteGraphBuilder** (unidad sgcMaps_OfflineRouter) lee datos de OpenStreetMap, un archivo .osm.pbf o .osm, y escribe un **grafo de rutas** binario compacto, un archivo .sgcgraph con los perfiles de coche, bicicleta y a pie.
2. TsgcMapOfflineRouter carga el grafo, ajusta cada punto de paso a la vía más cercana del perfil seleccionado y encuentra la ruta con una búsqueda A* bidireccional.

Creación del grafo

Build(archivo de entrada, archivo de salida) se ejecuta en tres fases, que informa **Phase**: 1 lee las vías, 2 lee las coordenadas de los nodos que usan esas vías, y 3 escribe el grafo. **OnProgress** informa de los bytes leídos y del tamaño total de la entrada mientras se lee. Cuando termina, WayCount, NodeCount, MissingNodes, VertexCount, EdgeCount, ShapeCount y NameCount describen el grafo. El grafo almacena de cada vía su longitud, su velocidad para coches, los sentidos en que se puede recorrer para cada perfil, sus puntos intermedios y su nombre.

```
procedure TForm2.BuildGraph;
var
  oBuilder: TsgcMapRouteGraphBuilder;
begin
  oBuilder := TsgcMapRouteGraphBuilder.Create;
  try
    oBuilder.OnProgress := GraphProgress;
    oBuilder.Build('andorra-latest.osm.pbf', 'andorra.sgcgraph');
    memoLog.Lines.Add(Format('%d ways, %d vertices, %d edges',
      [oBuilder.WayCount, oBuilder.VertexCount, oBuilder.EdgeCount]));
  finally
    oBuilder.Free;
  end;
end;

procedure TForm2.GraphProgress(Sender: TObject; aBytesRead, aTotal: Int64);
begin
  if aTotal > 0 then
    lblProgress.Caption := Format('Phase %d of 3: %d%%',
      [TsgcMapRouteGraphBuilder(Sender).Phase, aBytesRead * 100 div aTotal]);
  Application.ProcessMessages;
end;
```

Cree el grafo una sola vez, para la región que cubre la aplicación, y distribuya el archivo .sgcgraph. La herramienta de consola RouteGraphBuilder hace lo mismo desde la línea de comandos.

Cálculo de rutas

Establezca GraphFile y Profile (morCar, morBike o morFoot) y llame a Open. **GetRoute** recibe dos o más puntos de paso y rellena un **TsgcMapRoute**, la clase que usa el TsgcMapDirections en línea, con la geometría, la distancia, la duración, un tramo por cada par de puntos de paso y los pasos de giro. **sgcMapAddRouteToPolylines** la dibuja en un mapa.

```
procedure TForm2.RouteByBike;
var
  vWaypoints: array [0 .. 2] of TsgcMapCoord;
  oRoute: TsgcMapRoute;
begin
  sgcMapOfflineRouter1.GraphFile := 'andorra.sgcgraph';
  sgcMapOfflineRouter1.Profile := morBike;
  sgcMapOfflineRouter1.Open;
```

```
  vWaypoints[0] := sgcMapCoord(42.5063, 1.5218); // Andorra la Vella
  vWaypoints[1] := sgcMapCoord(42.5448, 1.5146); // La Massana
  vWaypoints[2] := sgcMapCoord(42.5347, 1.5800); // Encamp
  oRoute := TsgcMapRoute.Create;
  try
    if sgcMapOfflineRouter1.GetRoute(vWaypoints, oRoute) then
    begin
      sgcMapAddRouteToPolylines(oRoute, sgcVCLMap1.Polylines);
      memoLog.Lines.Add(Format('%.1f km in %d legs',
        [oRoute.Distance / 1000, oRoute.LegCount]));
    end
    else
      memoLog.Lines.Add(sgcMapOfflineRouter1.LastError);
  finally
    oRoute.Free;
  end;
end;
```

El enrutador sin conexión también puede ser el motor de TsgcMapRouteCalculator, a través de su evento OnCalculateRoute, lo que da al usuario puntos de paso arrastrables sin conexión de red.

Disponibilidad

Framework	Versión mínima
VCL, FireMonkey, consola y servicios	Delphi XE2
Lazarus	No disponible

Véase también

- Lector de OpenStreetMap
- TsgcMapOfflineRouter
- TsgcMapRouteCalculator

TsgcMapService

Unidad: sgcMaps_Services
La clase base de todos los servicios de mapas en línea: TsgcMapGeocoder, TsgcMapDirections, TsgcMapPlaces, TsgcMapElevation y TsgcMapTimeZone. No está registrada en la paleta. Los servicios envían solicitudes HTTPS con THTTPClient y analizan las respuestas con System.JSON, así que necesitan Delphi XE8 o posterior y no están disponibles en Lazarus.

Llamadas síncronas y asíncronas

- Los métodos **síncronos** (Geocode, GetRoute, Suggest...) bloquean el hilo que los llama hasta que llega la respuesta. Nunca generan excepciones: si fallan, devuelven False y establecen **LastError**.
- Los métodos **asíncronos** (GeocodeAsync, GetRouteAsync, SuggestAsync...) ejecutan la solicitud en un hilo en segundo plano y disparan el evento de resultado, u **OnError**, en el hilo principal mediante TThread.Queue. Las aplicaciones VCL y FireMonkey no necesitan nada más. Una aplicación de consola o un servicio debe llamar a CheckSynchronize periódicamente, o usar los métodos síncronos.
- El componente debe seguir existiendo hasta que una solicitud asíncrona haya disparado su evento. Los objetos de resultado que se pasan a los eventos pertenecen al componente y se liberan cuando el evento termina, así que copie lo que necesite conservar.

Servidores públicos y políticas de uso

Los proveedores predeterminados son servicios públicos gratuitos con políticas de uso. Nominatim, el servidor de demostración de OSRM y la API pública de OpenTopoData permiten como máximo una solicitud por segundo, y los componentes hacen cumplir ese intervalo. También esperan un **UserAgent** que los identifique. Use su propio servidor o un proveedor comercial para producción y para solicitudes masivas.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
ApiKey	string		Clave de los proveedores que la necesitan. Para el proveedor de zona horaria GeoNames es el nombre de usuario de GeoNames.
Language	string	en	Idioma de los resultados, para los proveedores que lo admiten.
Timeout	Integer	15000	Tiempo de espera de la solicitud en milisegundos.
UserAgent	string	sgcMaps/2.0	Cabecera User-Agent de cada solicitud. Identifique aquí su aplicación.
Referer	string		Cabecera Referer, que se envía cuando no está vacía.
LastError	string		Error de la última solicitud fallida. Solo lectura.

Eventos

Evento	Descripción

OnError	Falló una solicitud asíncrona. Se dispara en el hilo principal con el texto del error.

Ejemplo: configuración común de los servicios

Un procedimiento configura todos los servicios de un formulario, ya que las propiedades se declaran en la clase base.

```
procedure TForm1.SetupService(aService: TsgcMapService);
begin
  // public servers such as Nominatim require an identifying User-Agent
  aService.UserAgent := 'MyMapApp/1.0 (contact@example.com)';
  aService.Language := 'en';
  aService.Timeout := 10000;
  aService.OnError := ServiceError;
end;

procedure TForm1.ServiceError(Sender: TObject; const aError: string);
begin
  memoLog.Lines.Add(Sender.ClassName + ': ' + aError);
end;
```

Véase también

- TsgcMapGeocoder
- TsgcMapDirections
- TsgcMapPlaces
- TsgcMapElevation
- TsgcMapTimeZone

TsgcMapGeocoder

Unidad: sgcMaps_Geocoding
Convierte una dirección o un nombre de lugar en coordenadas (geocodificación), y unas coordenadas en una dirección (geocodificación inversa).
El componente desciende de TsgcMapService y hereda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** y **OnError**. Necesita Delphi XE8 o posterior. Los métodos síncronos nunca generan excepciones: devuelven False y establecen LastError. Los métodos asíncronos disparan sus eventos en el hilo principal.

Proveedores

Proveedor	Servicio	Clave
mgpNominatim	Nominatim (OpenStreetMap). Es el predeterminado. NominatimURL apunta a una instancia propia.	No
mgpGoogle	Google Geocoding API	Sí
mgpHere	HERE Geocoding and Search	Sí
mgpAzure	Azure Maps Search	Sí
mgpMapBox	Mapbox Geocoding	Sí
mgpOpenRouteService	OpenRouteService Geocode	Sí
mgpGeoApify	Geoapify Geocoding	Sí

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Provider	TsgcMapGeocodingProvider	mgpNominatim	El servicio de geocodificación.
NominatimURL	string	https://nominatim.openstreetmap.org	URL base del servidor Nominatim.
MaxResults	Integer	5	Número máximo de resultados.
CountryCodes	string		Filtro de países separados por comas con códigos ISO 3166-1 alfa-2 ('es,fr'). HERE espera códigos alfa-3 ('ESP,FRA') y Azure lo ignora.

Métodos

Método	Descripción

Geocode(const aQuery: string; aResults: TsgcMapGeocodingResults): Boolean	Busca una dirección o un lugar. True cuando se encontró al menos un resultado.
ReverseGeocode(aLat, aLon: Double; aResults: TsgcMapGeocodingResults): Boolean	Encuentra la dirección de una coordenada.
GeocodeAsync(const aQuery: string)	Geocode asíncrono. Los resultados llegan en OnGeocodeResult.
ReverseGeocodeAsync(aLat, aLon: Double)	ReverseGeocode asíncrono. Los resultados llegan en OnGeocodeResult.
BuildURL, BuildReverseURL	La URL de la solicitud para el proveedor actual, útil para registros y pruebas.

Eventos

Evento	Descripción
OnGeocodeResult	Los resultados de una solicitud asíncrona, en el hilo principal. La lista se libera cuando el evento termina.

Resultados

Cada resultado es un TsgcMapGeocodingResult con **Latitude**, **Longitude**, **DisplayName**, **Street**, **HouseNumber**, **PostalCode**, **City**, **State**, **Country**, **CountryCode** (ISO alfa-2 cuando el proveedor lo devuelve, alfa-3 para HERE), **Confidence** (de 0 a 1, 0 cuando se desconoce) y el cuadro delimitador **BoundsNorth**, **BoundsSouth**, **BoundsEast** y **BoundsWest** (véase HasBounds). Las mismas clases de resultado las usa TsgcMapOfflineGeocoder.

Ejemplo: geocodificación y geocodificación inversa

```
procedure TForm1.btnGeocodeClick(Sender: TObject);
var
  oResults: TsgcMapGeocodingResults;
begin
  sgcMapGeocoder1.Provider := mgpNominatim;
  sgcMapGeocoder1.MaxResults := 3;
  oResults := TsgcMapGeocodingResults.Create;
  try
    if sgcMapGeocoder1.Geocode('Plaza Mayor, Madrid', oResults) then
    begin
      sgcVCLMap1.SetCenter(oResults[0].Latitude, oResults[0].Longitude);
      sgcVCLMap1.AddMarker(oResults[0].Latitude, oResults[0].Longitude,
        oResults[0].DisplayName);
    end
    else
      memoLog.Lines.Add('Geocode failed: ' + sgcMapGeocoder1.LastError);
  finally
    oResults.Free;
  end;
end;

procedure TForm1.btnReverseClick(Sender: TObject);
begin
  sgcMapGeocoder1.OnGeocodeResult := sgcMapGeocoder1GeocodeResult;
  sgcMapGeocoder1.ReverseGeocodeAsync(sgcVCLMap1.CenterLatitude,
    sgcVCLMap1.CenterLongitude);
end;
```

```
procedure TForm1.sgcMapGeocoder1GeocodeResult(Sender: TObject;
  aResults: TsgcMapGeocodingResults);
begin
  // aResults is owned by the component: copy what you need to keep
  if aResults.Count > 0 then
    memoLog.Lines.Add(aResults[0].DisplayName);
end;
```

Véase también

- TsgcMapService
- TsgcMapPlaces
- TsgcMapOfflineGeocoder

TsgcMapDirections

Unidad: sgcMaps_Directions
Calcula una ruta a través de dos o más puntos de paso con un servicio de rutas en línea. Las distancias están en metros y las duraciones en segundos.
El componente desciende de TsgcMapService y hereda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** y **OnError**. Necesita Delphi XE8 o posterior. Los métodos síncronos nunca generan excepciones: devuelven False y establecen LastError. Los métodos asíncronos disparan sus eventos en el hilo principal.

Proveedores

Proveedor	Servicio	Clave
mdpOSRM	OSRM. El OSRMURL predeterminado es el servidor de demostración público del proyecto OSRM, que solo permite un uso ligero y ofrece únicamente el perfil de coche. Use su propio osrm-routed para producción y para rutas a pie y en bicicleta.	No
mdpOpenRouteService	OpenRouteService Directions	Sí
mdpGoogleRoutes	Google Routes API	Sí
mdpHere	HERE Routing	Sí
mdpAzure	Azure Maps Route	Sí
mdpMapBox	Mapbox Directions	Sí
mdpGeoApify	Geoapify Routing	Sí

Todos los proveedores devuelven las instrucciones paso a paso en **Language**, salvo OSRM, que no devuelve ninguna: para OSRM el componente crea instrucciones breves en inglés.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Provider	TsgcMapDirectionsProvider	mdpOSRM	El servicio de rutas.
OSRMURL	string	https://router.project-osrm.org	URL base del servidor OSRM.
TravelMode	TsgcMapTravelMode	mtmDriving	mtmDriving, mtmWalking, mtmCycling o mtmTruck. IsTravelModeSupported indica qué modos acepta un proveedor.
Alternatives	Boolean	False	Solicita rutas alternativas.

AvoidTolls, AvoidHighways, AvoidFerries	Boolean	False	Restricciones de la ruta, para los proveedores que las admiten.
OptimizeWaypoints	Boolean	False	Permite que el proveedor reordene los puntos de paso intermedios.

Métodos

Método	Descripción
GetRoute(const aWaypoints: array of TsgcMapCoord; aRoutes: TsgcMapRoutes): Boolean	Calcula la ruta. True cuando se encontró al menos una ruta; el índice 0 es la ruta principal.
GetRouteAsync(const aWaypoints: array of TsgcMapCoord)	GetRoute asíncrono. Las rutas llegan en OnRouteResult.
ValidateRequest(const aWaypoints: array of TsgcMapCoord): string	Una cadena vacía cuando se puede enviar la solicitud, y en caso contrario el motivo por el que no se puede.
IsTravelModeSupported(aProvider; aMode): Boolean	Función de clase: indica si un proveedor admite un modo de viaje.

Eventos

Evento	Descripción
OnRouteResult	Las rutas de una solicitud asíncrona, en el hilo principal. La lista se libera cuando el evento termina.

Resultados de ruta

Clase	Miembros
TsgcMapRoutes	Count e Items[Index] (propiedad predeterminada) de TsgcMapRoute.
TsgcMapRoute	Distance, Duration, Summary, la geometría (Points, PointCount, GetPoints), los tramos (Legs, LegCount) y el cuadro delimitador (BoundsNorth, BoundsSouth, BoundsEast, BoundsWest, CalcBounds).
TsgcMapRouteLeg	La parte de la ruta entre dos puntos de paso consecutivos: Distance, Duration, Summary, Steps y StepCount.
TsgcMapRouteStep	Una maniobra: Instruction, Maneuver, StreetName, Distance, Duration, Latitude, Longitude y PointIndex (su posición en la geometría de la ruta).

sgcMapAddRouteToPolylines(aRoute, aPolylines, aColor, aWidth) añade a un mapa una polilínea con la geometría de una ruta y la devuelve.

Ejemplo: calcular una ruta

```
procedure TForm1.btnRouteClick(Sender: TObject);
var
  vWaypoints: array [0 .. 1] of TsgcMapCoord;
  oRoutes: TsgcMapRoutes;
  oLine: TsgcMapPolyline;
  i, j: Integer;
begin
  vWaypoints[0].Latitude := 40.4168;
  vWaypoints[0].Longitude := -3.7038;
  vWaypoints[1].Latitude := 40.4530;
  vWaypoints[1].Longitude := -3.6883;

  sgcMapDirections1.Provider := mdpOSRM;
  sgcMapDirections1.TravelMode := mtmDriving;
  oRoutes := TsgcMapRoutes.Create;
  try
    if sgcMapDirections1.GetRoute(vWaypoints, oRoutes) then
    begin
      oLine := sgcMapAddRouteToPolylines(oRoutes[0], sgcVCLMap1.Polylines);
      sgcVCLMap1.FitToPolyline(oLine);
      memoLog.Lines.Add(Format('%.1f km, %.0f min',
        [oRoutes[0].Distance / 1000, oRoutes[0].Duration / 60]));
      for i := 0 to oRoutes[0].LegCount - 1 do
        for j := 0 to oRoutes[0].Legs[i].StepCount - 1 do
          memoLog.Lines.Add(oRoutes[0].Legs[i].Steps[j].Instruction);
    end
    else
      memoLog.Lines.Add('Route failed: ' + sgcMapDirections1.LastError);
  finally
    oRoutes.Free;
  end;
end;
```

Véase también

- TsgcMapService
- TsgcMapRouteCalculator
- TsgcMapOfflineRouter

TsgcMapPlaces

Unidad: sgcMaps_Places
Autocompletado de lugares: sugerencias para el texto que escribe el usuario, normalmente mostradas en una lista bajo un cuadro de búsqueda.
El componente desciende de TsgcMapService y hereda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** y **OnError**. Necesita Delphi XE8 o posterior. Los métodos síncronos nunca generan excepciones: devuelven False y establecen LastError. Los métodos asíncronos disparan sus eventos en el hilo principal.

Proveedores

Proveedor	Servicio	Clave
mplPhoton	Photon (komoot). Es el predeterminado. Solo acepta los idiomas de, en y fr; cualquier otro Language no se envía.	No
mplNominatim	Nominatim. Limitado a una solicitud por segundo, así que no use el servidor público para el autocompletado a cada pulsación de tecla.	No
mplGoogle	Google Places API (New). Las sugerencias no tienen coordenadas: llame a Retrieve.	Sí
mplHere	HERE Autosuggest	Sí
mplMapBox	Mapbox Search Box API. Las sugerencias no tienen coordenadas: llame a Retrieve.	Sí
mplGeoApify	Geoapify Autocomplete	Sí

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Provider	TsgcMapPlacesProvider	mplPhoton	El servicio de autocompletado.
NominatimURL	string	https://nominatim.openstreetmap.org	URL base del servidor Nominatim.
MaxResults	Integer	5	Número máximo de sugerencias.
BiasLatitude, BiasLongitude	Double	0	Se prefieren los resultados cercanos a esta posición y se rellena su Distance. No se usa cuando ambos son 0.
BiasRadius	Double	0	Radio alrededor de la posición de sesgo, en metros. 0 significa sin radio. Lo usan Google, HERE y Nominatim.
SessionToken	string		Token de la sesión de búsqueda actual. Público.

Métodos

Método	Descripción
Suggest(const aText: string; aResults: TsgcMapPlaceResults): Boolean	Devuelve las sugerencias para un texto. True cuando se encontró al menos una.
SuggestAsync(const aText: string)	Suggest asíncrono. Las sugerencias llegan en OnSuggestResult.
Retrieve(aPlace: TsgcMapPlace): Boolean	Rellena las coordenadas, y el nombre o la dirección que falten, de una sugerencia. Solo Google y Mapbox hacen una solicitud; para los demás proveedores devuelve HasCoordinates.
NewSession	Inicia una nueva sesión de búsqueda con un SessionToken nuevo.

Eventos

Evento	Descripción
OnSuggestResult	Las sugerencias de una solicitud asíncrona, en el hilo principal. La lista se libera cuando el evento termina.

Resultados

Cada sugerencia es un TsgcMapPlace con **Id** (el identificador del proveedor, que usa Retrieve), **Name**, **Address**, **Latitude**, **Longitude**, **HasCoordinates**, **Category** y **Distance** (metros desde la posición de sesgo, -1 cuando se desconoce).

Ejemplo: sugerencias mientras se escribe

```
procedure TForm1.edtSearchChange(Sender: TObject);
begin
  if Length(edtSearch.Text) < 3 then
    Exit;
  // prefer the places near the center of the map
  sgcMapPlaces1.BiasLatitude := sgcVCLMap1.CenterLatitude;
  sgcMapPlaces1.BiasLongitude := sgcVCLMap1.CenterLongitude;
  sgcMapPlaces1.SuggestAsync(edtSearch.Text);
end;

procedure TForm1.sgcMapPlaces1SuggestResult(Sender: TObject;
  aResults: TsgcMapPlaceResults);
var
  i: Integer;
begin
  lbPlaces.Items.Clear;
  for i := 0 to aResults.Count - 1 do
    lbPlaces.Items.Add(aResults[i].Name + ', ' + aResults[i].Address);
end;
```

Véase también

- TsgcMapService
- TsgcMapGeocoder

TsgcMapElevation

Unidad: sgcMaps_Elevation
Devuelve la elevación del terreno de una lista de puntos, o de muestras equiespaciadas a lo largo de un trayecto, por ejemplo un perfil de elevación de una ruta. Las listas largas se dividen en lotes automáticamente.
El componente desciende de TsgcMapService y hereda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** y **OnError**. Necesita Delphi XE8 o posterior. Los métodos síncronos nunca generan excepciones: devuelven False y establecen LastError. Los métodos asíncronos disparan sus eventos en el hilo principal.

Proveedores

Proveedor	Servicio	Puntos por solicitud	Clave
meOpenElevation	Open-Elevation	200	No
meOpenTopoData	OpenTopoData. Es el predeterminado. La API pública está limitada a una solicitud por segundo.	100	No
meGoogle	Google Elevation API	256	Sí

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Provider	TsgcMapElevationProvider	meOpenTopoData	El servicio de elevación.
Dataset	string	srtm90m	Conjunto de datos de OpenTopoData: srtm90m, srtm30m, aster30m, eudem25m y otros.

Métodos

Método	Descripción
GetElevations(const aPoints: array of TsgcMapCoord; aResults: TsgcMapElevationResults): Boolean	Un resultado por punto, en el mismo orden.
GetElevationAlongPath(const aPoints: array of TsgcMapCoord; aSamples: Integer; aResults: TsgcMapElevationResults): Boolean	aSamples puntos (al menos 2, incluidos el primero y el último) equiespaciados por distancia a lo largo de la polilínea aPoints.
GetElevationsAsync, GetElevationAlongPathAsync	Versiones asíncronas. Los resultados llegan en OnElevationResult.
BatchSize: Integer	Máximo de puntos por solicitud del proveedor actual.
SamplePath	Función de clase: las muestras equiespaciadas de una polilínea y sus distancias desde el primer punto.

Eventos

Evento	Descripción
OnElevationResult	Los resultados de una solicitud asíncrona, en el hilo principal. La lista se libera cuando el evento termina.

Resultados

Cada resultado es un TsgcMapElevationResult con **Latitude**, **Longitude**, **Elevation** (metros sobre el nivel del mar), **HasElevation** (False cuando el proveedor no tiene datos para el punto) y **Distance** (metros desde el primer punto, que rellena GetElevationAlongPath).

Ejemplo: perfil de elevación

```
procedure TForm1.btnElevationClick(Sender: TObject);
var
  vPath: array [0 .. 1] of TsgcMapCoord;
  oResults: TsgcMapElevationResults;
  i: Integer;
begin
  vPath[0].Latitude := 40.4168; // Madrid
  vPath[0].Longitude := -3.7038;
  vPath[1].Latitude := 40.7934; // Navacerrada pass
  vPath[1].Longitude := -4.0102;

  sgcMapElevation1.Provider := meOpenTopoData;
  sgcMapElevation1.Dataset := 'srtm90m';
  oResults := TsgcMapElevationResults.Create;
  try
    if not sgcMapElevation1.GetElevationAlongPath(vPath, 20, oResults) then
      memoLog.Lines.Add('Elevation failed: ' + sgcMapElevation1.LastError)
    else
      for i := 0 to oResults.Count - 1 do
        if oResults[i].HasElevation then
          memoLog.Lines.Add(Format('%.0f m along the path: %.0f m',
            [oResults[i].Distance, oResults[i].Elevation]));
  finally
    oResults.Free;
  end;
end;
```

Véase también

- TsgcMapService
- TsgcMapDirections

TsgcMapTimeZone

Unidad: sgcMaps_TimeZone
Devuelve la zona horaria de una coordenada: su identificador IANA, el desfase respecto a UTC y la hora local.
El componente desciende de TsgcMapService y hereda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** y **OnError**. Necesita Delphi XE8 o posterior. Los métodos síncronos nunca generan excepciones: devuelven False y establecen LastError. Los métodos asíncronos disparan sus eventos en el hilo principal.

Proveedores

Proveedor	Servicio	Clave
mtzTimeAPI	TimeAPI (timeapi.io). Es el predeterminado.	No
mtzGeoNames	GeoNames. ApiKey es el nombre de usuario de GeoNames.	Nombre de usuario
mtzGoogle	Google Time Zone API	Sí
mtzAzure	Azure Maps Timezone	Sí

El parámetro opcional **aUtcTime** (0 significa ahora) lo respetan Google y Azure. TimeAPI y GeoNames siempre responden para la hora actual.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Provider	TsgcMapTimeZoneProvider	mtzTimeAPI	El servicio de zona horaria.

Métodos

Método	Descripción
GetTimeZone(aLat, aLon: Double; aInfo: TsgcMapTimeZoneInfo; aUtcTime: TDateTime = 0): Boolean	Rellena aInfo con la zona horaria de la coordenada.
GetTimeZoneAsync(aLat, aLon: Double; aUtcTime: TDateTime = 0)	GetTimeZone asíncrono. La respuesta llega en OnTimeZoneResult.
UtcNow: TDateTime	Función de clase: la hora UTC actual.

Eventos

Evento	Descripción
OnTimeZoneResult	La respuesta de una solicitud asíncrona, en el hilo principal. El objeto se libera cuando el evento termina.

Resultados

TsgcMapTimeZoneInfo tiene **TimeZoneId** ('Europe/Madrid'), **Name** (el nombre para mostrar cuando el proveedor devuelve uno, y en caso contrario el TimeZoneId), **UtcOffsetSeconds** (estándar más horario de verano), **DstOffsetSeconds** (la parte del horario de verano), **IsDst** y **LocalTime** (ISO 8601 sin desfase).

Ejemplo: zona horaria de una coordenada

```
procedure TForm1.btnTimeZoneClick(Sender: TObject);
var
  oInfo: TsgcMapTimeZoneInfo;
begin
  sgcMapTimeZone1.Provider := mtzTimeAPI;
  oInfo := TsgcMapTimeZoneInfo.Create;
  try
    if sgcMapTimeZone1.GetTimeZone(35.6762, 139.6503, oInfo) then
      memoLog.Lines.Add(Format('%s, UTC offset %d s, local time %s',
        [oInfo.TimeZoneId, oInfo.UtcOffsetSeconds, oInfo.LocalTime]))
    else
      memoLog.Lines.Add('Time zone failed: ' + sgcMapTimeZone1.LastError);
  finally
    oInfo.Free;
  end;
end;
```

Véase también

- TsgcMapService

TsgcMapRouteCalculator

Unidad: sgcMaps_RouteCalculator
Edición interactiva de rutas en un mapa. Los puntos de paso son marcadores arrastrables en **Markers** y la ruta es una polilínea en **Polylines**; apunte ambos a las colecciones de un control de mapa. Arrastrar un punto de paso, añadir uno o eliminar uno recalcula la ruta cuando **AutoCalculate** es True. Necesita Delphi XE2 o posterior; la propiedad Directions necesita Delphi XE8.
El motor de rutas es el evento **OnCalculateRoute**, donde cualquier motor puede rellenar la ruta (por ejemplo TsgcMapOfflineRouter), o, cuando el evento no la gestiona, el TsgcMapDirections en línea asignado a **Directions**.
Los marcadores de puntos de paso reciben Tag = MarkerTagBase + índice y la polilínea de la ruta recibe Tag = MarkerTagBase. Con ShowAlternatives las rutas alternativas se dibujan como polilíneas grises más finas con Tag = MarkerTagBase + 1.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Markers	TsgcMapMarkers		Colección de los marcadores de puntos de paso, por ejemplo Map.Markers.
Polylines	TsgcMapPolylines		Colección de las polilíneas de la ruta, por ejemplo Map.Polylines.
Directions	TsgcMapDirections		Motor de rutas en línea, que se usa cuando OnCalculateRoute no gestiona la solicitud.
Active	Boolean	True	False desconecta el calculador de Markers y Polylines: sin seguimiento de los arrastres de marcadores ni cálculo automático.
AutoCalculate	Boolean	True	Recalcula después de cada cambio de los puntos de paso.
ShowAlternatives	Boolean	False	Dibuja también las rutas alternativas.
RouteColor, RouteWidth	TColor, Integer	azul, 4	Color y anchura de la polilínea de la ruta.
WaypointColor	TColor	red	Color de los marcadores de puntos de paso.
MarkerTagBase	Integer	SGC_MAP_ROUTE_TAG_BASE	Primer Tag que usan los marcadores y las polilíneas del calculador.
Waypoints[Index], WaypointMarkers[Index]	TsgcMapCoord, TsgcMapMarker		Los puntos de paso y sus marcadores. Solo lectura.

Route, RoutePolyline, Routes	TsgcMapRoute, TsgcMapPolyline, TsgcMapRoutes		La ruta principal, su polilínea y todas las rutas del último cálculo (el índice 0 es la ruta principal). Solo lectura.

Métodos

Método	Descripción
AddWaypoint(aLat, aLon: Double): Integer	Añade un punto de paso al final y devuelve su índice.
InsertWaypoint(aIndex: Integer; aLat, aLon: Double)	Inserta un punto de paso.
AddWaypointNearestSegment(aLat, aLon: Double): Integer	Inserta un punto de paso entre los dos puntos de paso cuyo tramo está más cerca del punto, como hace arrastrar la línea de la ruta.
MoveWaypoint(aIndex: Integer; aLat, aLon: Double)	Mueve un punto de paso.
RemoveWaypoint(aIndex: Integer), ClearWaypoints	Elimina un punto de paso o todos.
WaypointCount: Integer	Número de puntos de paso.
Calculate: Boolean	Cálculo síncrono: OnCalculateRoute y después Directions.GetRoute.
CalculateAsync	OnCalculateRoute, aplicado al instante cuando se gestiona, y en caso contrario Directions.GetRouteAsync. Los eventos se disparan en el hilo principal.
SelectRoute(aIndex: Integer)	Convierte una ruta alternativa en la ruta principal, la vuelve a dibujar y dispara OnRouteCalculated.

Eventos

Evento	Descripción
OnCalculateRoute	Se dispara primero en cada cálculo. Rellene aRoute y establezca aHandled en True para usar su propio motor.
OnRouteCalculated	Se calculó y se dibujó una ruta nueva.
OnWaypointsChanged	Cambiaron los puntos de paso.
OnError	Falló el cálculo.

Ejemplo: crear una ruta haciendo clic en el mapa

```
procedure TForm1.SetupRouteCalculator;
begin
  // the calculator draws on the collections of the map
  sgcMapRouteCalculator1.Markers := sgcVCLMap1.Markers;
  sgcMapRouteCalculator1.Polylines := sgcVCLMap1.Polylines;
  sgcMapRouteCalculator1.Directions := sgcMapDirections1;
  sgcMapRouteCalculator1.AutoCalculate := True;
  sgcMapRouteCalculator1.OnRouteCalculated :=
    sgcMapRouteCalculator1RouteCalculated;
  sgcVCLMap1.OnMapClick := MapClickAddWaypoint;
end;

procedure TForm1.MapClickAddWaypoint(Sender: TObject; aLat, aLon: Double);
begin
  sgcMapRouteCalculator1.AddWaypoint(aLat, aLon);
end;

procedure TForm1.sgcMapRouteCalculator1RouteCalculated(Sender: TObject;
  aRoute: TsgcMapRoute);
begin
  memoLog.Lines.Add(Format('Route: %.1f km, %.0f min',
    [aRoute.Distance / 1000, aRoute.Duration / 60]));
end;
```

Para calcular la ruta sin conexión de red, gestione OnCalculateRoute con el enrutador sin conexión, como se muestra en TsgcMapOfflineRouter.

Véase también

- TsgcMapDirections
- TsgcMapOfflineRouter
- TsgcVCLMap

TsgcMapOfflineRouter

Unidad: sgcMaps_OfflineRouter
Calcula rutas sin conexión de red, sobre un archivo de grafo de rutas (**.sgcgraph**) creado a partir de datos de OpenStreetMap. Necesita Delphi XE2 o posterior.
Cree el grafo una sola vez a partir de un extracto de OpenStreetMap (.osm.pbf, o XML .osm) con **TsgcMapRouteGraphBuilder**, declarado en la misma unidad, o con la herramienta RouteGraphBuilder de Tools\Maps. El enrutador rellena la misma clase TsgcMapRoute que TsgcMapDirections, así que una ruta de cualquiera de los dos motores se dibuja y se inspecciona del mismo modo. Una ruta a través de varios puntos de paso tiene un tramo por cada par de puntos de paso consecutivos.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
GraphFile	string		El archivo .sgcgraph.
Profile	TsgcMapOfflineRouterProfile	morCar	morCar, morBike o morFoot.
Active	Boolean	False	Abre o cierra el grafo.
LastError	string		Error de la última llamada fallida. Solo lectura.
VertexCount, EdgeCount	Integer		Tamaño del grafo cargado. Solo lectura.

Métodos

Método	Descripción
Open, Close	Carga o libera el grafo.
GetRoute(const aWaypoints: array of TsgcMapCoord; aRoute: TsgcMapRoute): Boolean	Calcula la ruta. False cuando no se encontró ninguna ruta, con el motivo en LastError.

Ejemplo: ruta sin conexión y motor sin conexión del calculador de rutas

```
procedure TForm1.btnOfflineRouteClick(Sender: TObject);
var
  vWaypoints: array [0 .. 1] of TsgcMapCoord;
  oRoute: TsgcMapRoute;
begin
  sgcMapOfflineRouter1.GraphFile := 'andorra.sgcgraph';
  sgcMapOfflineRouter1.Profile := morCar;
  sgcMapOfflineRouter1.Open;

  vWaypoints[0].Latitude := 42.5063; // Andorra la Vella
  vWaypoints[0].Longitude := 1.5218;
  vWaypoints[1].Latitude := 42.5347; // Encamp
  vWaypoints[1].Longitude := 1.5800;
  oRoute := TsgcMapRoute.Create;
```

```
  try
    if sgcMapOfflineRouter1.GetRoute(vWaypoints, oRoute) then
      sgcMapAddRouteToPolylines(oRoute, sgcVCLMap1.Polylines)
    else
      memoLog.Lines.Add('No route: ' + sgcMapOfflineRouter1.LastError);
  finally
    oRoute.Free;
  end;
end;

procedure TForm1.sgcMapRouteCalculator1CalculateRoute(Sender: TObject;
  const aWaypoints: array of TsgcMapCoord; aRoute: TsgcMapRoute;
  var aHandled: Boolean);
begin
  // the offline router as the engine of the interactive route calculator
  aHandled := sgcMapOfflineRouter1.GetRoute(aWaypoints, aRoute);
end;
```

Véase también

- TsgcMapRouteCalculator
- TsgcMapDirections

TsgcMapOfflineGeocoder

Unidad: sgcMaps_OfflineGeocoder
Geocodificación y geocodificación inversa sin conexión de red, sobre un índice SQLite creado a partir de los volcados de GeoNames por TsgcMapGeoIndexBuilder. Usa el controlador FireDAC SQLite, así que necesita Delphi XE7 o posterior.
Las búsquedas directas aceptan un nombre de lugar, un prefijo y un país o una región opcionales: 'Madrid', 'mostol', 'Toledo, US', 'Madrid ES' o 'Toledo, Ohio'. Los resultados se ordenan por coincidencia exacta antes que por coincidencia de prefijo, después por población y después por código de característica (capital, después sede administrativa, después otros lugares poblados). Las búsquedas inversas devuelven los lugares más cercanos en un radio de 50 km, primero los lugares poblados, con cada grupo ordenado por distancia.
Cada componente tiene una conexión SQLite y debe usarse desde un solo hilo a la vez. Para consultar el mismo índice desde varios hilos, cree un componente por hilo; el archivo se abre en modo de solo lectura.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
DatabaseFile	string		El archivo de índice SQLite.
Active	Boolean	False	Abre o cierra el índice.
MaxResults	Integer	5	Número máximo de resultados.
CountryCodes	string		Códigos ISO alfa-2 separados por comas que restringen todas las búsquedas.
Language	string		Sugerencia de idioma. Los volcados de GeoNames no indican el idioma de cada nombre alternativo, así que por ahora el valor se ignora.

Métodos

Método	Descripción
Open, Close	Abre o cierra el índice.
Geocode(const aQuery: string; aResults: TsgcMapGeocodingResults): Boolean	Búsqueda directa. True cuando se encontró al menos un lugar.
ReverseGeocode(aLatitude, aLongitude: Double; aResults: TsgcMapGeocodingResults): Boolean	Los lugares más cercanos en un radio de 50 km.
ReverseGeocodeCountry(aLatitude, aLongitude: Double): string	Código ISO del país del lugar más cercano, vacío cuando no hay ninguno en un radio de 50 km.

Los resultados usan la misma clase TsgcMapGeocodingResult que TsgcMapGeocoder.

Ejemplo: búsqueda sin conexión

```
procedure TForm1.btnOfflineSearchClick(Sender: TObject);
var
  oResults: TsgcMapGeocodingResults;
  i: Integer;
begin
  sgcMapOfflineGeocoder1.DatabaseFile := 'geonames.db';
  sgcMapOfflineGeocoder1.MaxResults := 5;
  sgcMapOfflineGeocoder1.Open;
  oResults := TsgcMapGeocodingResults.Create;
  try
    if sgcMapOfflineGeocoder1.Geocode('Toledo, US', oResults) then
      for i := 0 to oResults.Count - 1 do
        memoLog.Lines.Add(Format('%s (%.4f, %.4f)', [oResults[i].DisplayName,
          oResults[i].Latitude, oResults[i].Longitude]));
    memoLog.Lines.Add('Country at the map center: ' +
      sgcMapOfflineGeocoder1.ReverseGeocodeCountry(sgcVCLMap1.CenterLatitude,
      sgcVCLMap1.CenterLongitude));
  finally
    oResults.Free;
  end;
end;
```

Véase también

- TsgcMapGeoIndexBuilder
- TsgcMapGeocoder

TsgcMapGeoIndexBuilder

Unidad: sgcMaps_OfflineGeocoder
Crea el índice SQLite en el que busca TsgcMapOfflineGeocoder, a partir de un volcado de GeoNames descargado de https://download.geonames.org/export/dump/: cities500.zip, cities1000.zip, allCountries.zip, un archivo de país como ES.zip, o el archivo .txt extraído. Necesita Delphi XE7 o posterior (FireDAC SQLite).
La creación se ejecuta en el hilo que llama. La herramienta GeoIndexBuilder de Tools\Maps envuelve este componente en una aplicación de consola.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
InputFile	string		El volcado de GeoNames: el archivo .txt separado por tabuladores o un .zip que lo contiene.
DatabaseFile	string		El archivo SQLite que se crea. Un archivo existente se sobrescribe.
Admin1CodesFile	string		admin1CodesASCII.txt opcional, para los nombres de las regiones de primer nivel.
CountryInfoFile	string		countryInfo.txt opcional, para los nombres de los países.
FeatureClasses	string	PA	Clases de características de GeoNames que se importan: P lugares poblados, A áreas administrativas, S puntos de interés, T terreno, L áreas, H hidrografía, R carreteras, U submarino, V vegetación.
MinPopulation	Int64	0	Los lugares poblados por debajo de esta población se omiten. Las demás clases no se filtran.
IncludeAlternateNames	Boolean	True	Indexa los nombres alternativos de hasta 64 caracteres, como los nombres cortos y los nombres en otros idiomas.
BatchSize	Integer	2000	Filas escritas por lote.
RowsRead, PlacesWritten, NamesWritten, ElapsedMs	Int64		Estadísticas de la última creación. Solo lectura.

Métodos

Método	Descripción
Build	Crea el índice.
RowsPerSecond: Double	Velocidad de la última creación.

Eventos

Evento	Descripción
OnProgress	Se dispara durante la creación con las filas leídas, los lugares escritos y los milisegundos transcurridos.

Ejemplo: crear un índice

```
procedure TForm1.btnBuildIndexClick(Sender: TObject);
begin
  sgcMapGeoIndexBuilder1.InputFile := 'cities500.zip';
  sgcMapGeoIndexBuilder1.Admin1CodesFile := 'admin1CodesASCII.txt';
  sgcMapGeoIndexBuilder1.CountryInfoFile := 'countryInfo.txt';
  sgcMapGeoIndexBuilder1.DatabaseFile := 'geonames.db';
  sgcMapGeoIndexBuilder1.FeatureClasses := 'PA';
  sgcMapGeoIndexBuilder1.OnProgress := sgcMapGeoIndexBuilder1Progress;
  sgcMapGeoIndexBuilder1.Build;
  memoLog.Lines.Add(Format('%d places indexed in %d ms',
    [sgcMapGeoIndexBuilder1.PlacesWritten, sgcMapGeoIndexBuilder1.ElapsedMs]));
end;

procedure TForm1.sgcMapGeoIndexBuilder1Progress(Sender: TObject;
  aRowsRead, aPlacesWritten, aElapsedMs: Int64);
begin
  Caption := Format('%d rows read, %d places', [aRowsRead, aPlacesWritten]);
  Update;
end;
```

Véase también

- TsgcMapOfflineGeocoder

TsgcMapLocation

Unidad: sgcMaps_Location
La posición del dispositivo. Envuelve el TLocationSensor de System.Sensors, así que necesita Delphi XE6 o posterior y funciona en aplicaciones VCL y FireMonkey: Windows usa la API de ubicación de Windows, y Android, iOS y macOS usan sus servicios nativos de ubicación.
El componente no conoce ningún mapa. **OnLocationChanged** informa de cada lectura, y **UpdateMarker** y **UpdateAccuracyCircle** mantienen un marcador de posición y un círculo de precisión en las colecciones de un mapa. Con **FollowMode** establecido en mfmCenter, cada lectura dispara además **OnFollow**, donde la aplicación centra su mapa.

Permisos

- Android: declare ACCESS_FINE_LOCATION en Uses Permissions de las opciones del proyecto, y solicítelo en tiempo de ejecución antes de establecer Active en True, con **RequestPermission** (Delphi 10.3 o posterior) o con PermissionsService.RequestPermissions.
- iOS y macOS: añada las claves de descripción de uso de la ubicación a la información de versión del proyecto.

Propiedades

Propiedad	Tipo	Predeterminado	Descripción
Active	Boolean	False	Inicia o detiene el sensor de ubicación.
Accuracy	Double		Precisión deseada del sensor, en metros.
Distance	Double		Movimiento mínimo, en metros, entre dos lecturas notificadas.
FollowMode	TsgcMapLocationFollowMode	mfmNone	mfmCenter dispara OnFollow después de cada lectura.
Available	Boolean		Indica si el dispositivo tiene un sensor de ubicación. Solo lectura.
HasLocation	Boolean		Indica si ha llegado una lectura. Solo lectura.
LastLatitude, LastLongitude, LastAccuracy, LastHeading, LastSpeed, LastTimestamp			La última lectura. Solo lectura.
TrackCount, TrackIndex, TrackIntervalMs, TrackAccuracy			Estado del recorrido simulado cargado con LoadTrack. Solo lectura.

Métodos

Método	Descripción

UpdateMarker(aMarkers: TsgcMapMarkers; aTag: Integer)	Añade o mueve el marcador de posición, identificado por su Tag.
UpdateAccuracyCircle(aCircles: TsgcMapCircles; aTag: Integer)	Añade o actualiza un círculo con el radio de la precisión.
RequestPermission	Solicita el permiso de ubicación en Android. La respuesta llega en OnPermissionResult.
SimulateLocation(aLatitude, aLongitude: Double; aAccuracy: Double = 0; aHeading: Double = -1; aSpeed: Double = -1)	Notifica una posición como si el sensor la hubiera leído, para pruebas.
LoadTrack(const aPoints: array of TsgcMapCoord; aIntervalMs: Integer = 1000; aAccuracy: Double = 10)	Carga una lista de puntos, por ejemplo los puntos de una ruta, para simular un trayecto.
StepTrack: Boolean	Notifica el siguiente punto del recorrido cargado. False al final del recorrido.
ResetTrack	Vuelve al primer punto del recorrido.

Eventos

Evento	Descripción
OnLocationChanged	Una lectura nueva: latitud, longitud, precisión, rumbo y velocidad.
OnFollow	Con FollowMode = mfmCenter, se dispara después de OnLocationChanged con la posición en la que centrar.
OnStateChanged	El sensor se inició o se detuvo.
OnError	El sensor notificó un error.
OnPermissionResult	La respuesta a RequestPermission.

Ejemplo: mostrar y seguir la posición

```
procedure TForm1.chkFollowClick(Sender: TObject);
begin
  if chkFollow.Checked then
    sgcMapLocation1.FollowMode := mfmCenter
  else
    sgcMapLocation1.FollowMode := mfmNone;
  sgcMapLocation1.Active := True;
end;

procedure TForm1.sgcMapLocation1LocationChanged(Sender: TObject;
  aLatitude, aLongitude, aAccuracy, aHeading, aSpeed: Double);
begin
  // keeps a "my location" marker and its accuracy circle on the map
  sgcMapLocation1.UpdateMarker(sgcVCLMap1.Markers);
  sgcMapLocation1.UpdateAccuracyCircle(sgcVCLMap1.Circles);
end;

procedure TForm1.sgcMapLocation1Follow(Sender: TObject;
  aLatitude, aLongitude: Double);
begin
```

```
  sgcVCLMap1.SetCenter(aLatitude, aLongitude);
end;

procedure TForm1.btnSimulateClick(Sender: TObject);
begin
  // a fixed position with an accuracy of 15 meters, for testing
  sgcMapLocation1.SimulateLocation(40.4168, -3.7038, 15);
end;
```

Véase también

- TsgcVCLMap
- TsgcFMXMap

Herramientas

La carpeta **Tools\Maps** contiene dos aplicaciones de consola que preparan los datos de los componentes sin conexión. Envuelven las clases constructoras de sgcMaps, de modo que el mismo trabajo se puede hacer desde su propio código.

Herramienta	Entrada	Salida	Lo usa	Versión mínima
GeoIndexBuilder	Un volcado de GeoNames	Un índice SQLite	TsgcMapOfflineGeocoder	Delphi XE7 (FireDAC SQLite)
RouteGraphBuilder	Un extracto de OpenStreetMap	Un archivo .sgcgraph	TsgcMapOfflineRouter	Delphi XE2

Compilación de las herramientas

Cada herramienta es un único archivo .dpr. Ábralo en el IDE y compílelo, o compílelo desde un símbolo del sistema con la carpeta Source en las rutas de unidades y de inclusión, por ejemplo:

```
dcc32 -NSSystem;Data;FireDAC -U..\..\..\Source -I..\..\..\Source sgcMapsGeoIndexBuilder.dpr
```

Cree los datos una sola vez, en una máquina de desarrollo, y distribuya los archivos resultantes con su aplicación.

GeoIndexBuilder

sgcMapsGeoIndexBuilder (Tools\Maps\GeoIndexBuilder) crea el índice SQLite del geocodificador sin conexión a partir de un volcado de GeoNames. Es un envoltorio de consola de TsgcMapGeoIndexBuilder y necesita Delphi XE7 o posterior, ya que escribe el índice con FireDAC SQLite.

Datos de entrada

Los volcados de GeoNames son descargas gratuitas de https://download.geonames.org/export/dump/:

- **cities500.zip**, **cities1000.zip**, **cities5000.zip** o **cities15000.zip**: los lugares poblados del mundo por encima de esa población.
- **XX.zip**, como ES.zip: todos los elementos de un país.
- **allCountries.zip**: todos los elementos del mundo.
- **admin1CodesASCII.txt** (opcional): los nombres de las áreas administrativas de primer nivel, como estados y regiones.
- **countryInfo.txt** (opcional): los nombres de los países, que se usan para completar búsquedas como "Paris, France".

La herramienta lee directamente el archivo .zip, o el archivo .txt extraído.

Línea de comandos

```
sgcMapsGeoIndexBuilder <input.zip|txt> <output.sqlite> [--minpop N]
  [--admin1 file] [--countryinfo file] [--classes PA] [--noalternates]
```

Parámetro	Propiedad	Significado
input	InputFile	El volcado de GeoNames, .zip o .txt.
output	DatabaseFile	El archivo SQLite que se crea. Un archivo existente se sobrescribe.
--minpop N	MinPopulation	Omite los lugares poblados (clase de característica P) por debajo de esta población. Las demás clases de características no se filtran.
--admin1 file	Admin1CodesFile	El archivo admin1CodesASCII.txt.
--countryinfo file	CountryInfoFile	El archivo countryInfo.txt.
--classes	FeatureClasses	Las clases de características de GeoNames que se importan. El valor predeterminado es PA: lugares poblados y áreas administrativas. Las demás son S (puntos de interés), T (terreno), L (áreas), H (hidrografía), R (carreteras), U (submarino) y V (vegetación).
--noalternates	IncludeAlternateNames	Omite los nombres alternativos (nombres cortos y nombres en otros idiomas), lo que reduce el tamaño del índice.

```
sgcMapsGeoIndexBuilder cities500.zip world.sqlite --admin1 admin1CodesASCII.txt --countryinfo countryInfo.txt
```

Salida

La herramienta muestra su progreso y después las filas leídas, los lugares y los nombres escritos, el tiempo y el tamaño del índice:

```
Building world.sqlite from cities500.zip
  ... rows read, ... places, ... s
Done: ... rows read, ... places, ... names, ... s, ... rows/s, ... MB
```

Termina con el código 0 si todo va bien, 1 si los parámetros son incorrectos y 2 cuando falla la creación, después de mostrar el error.

Uso del índice

Distribuya el archivo .sqlite con la aplicación y ábralo con TsgcMapOfflineGeocoder estableciendo su DatabaseFile. El componente responde entonces a Geocode, ReverseGeocode y ReverseGeocodeCountry sin conexión de red. La aplicación necesita el controlador FireDAC SQLite en tiempo de ejecución, como se describe en Solución de problemas. Índice de geocodificación sin conexión explica el contenido del índice.

RouteGraphBuilder

sgcMapsRouteGraphBuilder (Tools\Maps\RouteGraphBuilder) convierte un extracto de OpenStreetMap en el grafo de rutas del enrutador sin conexión. Es un envoltorio de consola de TsgcMapRouteGraphBuilder (unidad sgcMaps_OfflineRouter) y necesita Delphi XE2 o posterior.

Datos de entrada

Un extracto de OpenStreetMap en formato **.osm.pbf** o como XML **.osm**. Servicios de descarga como Geofabrik publican extractos de cada continente, país y región. Elija el extracto más pequeño que cubra el área de su aplicación, ya que el tiempo y la memoria de la creación crecen con el extracto.

Línea de comandos

```
sgcMapsRouteGraphBuilder <input.osm.pbf|input.osm> <output.sgcgraph>

sgcMapsRouteGraphBuilder andorra-latest.osm.pbf andorra.sgcgraph
```

Salida

El constructor lee el archivo dos veces, primero las vías y después las coordenadas de los nodos que usan esas vías, y muestra su progreso cada 10 por ciento. Al final muestra las estadísticas del grafo:

```
Input : andorra-latest.osm.pbf
Output: andorra.sgcgraph
  reading ways   0%
  ...
  reading nodes 100%
Routable ways : ...
Nodes used    : ... (missing ...)
Vertices      : ...
Edges         : ...
Shape points  : ...
Street names  : ...
Graph size    : ... bytes
Time          : ... ms
```

Nodes used cuenta los nodos a los que hacen referencia las vías transitables, y missing cuenta los que no están en el archivo, lo que ocurre con las vías que cruzan el borde de un extracto. La herramienta termina con el código 0 si todo va bien, 2 si los parámetros son incorrectos y 1 cuando falla la creación, después de mostrar el error.

Uso del grafo

Un único archivo .sgcgraph contiene los perfiles de coche, bicicleta y a pie. Distribúyalo con la aplicación, establézcalo como GraphFile de TsgcMapOfflineRouter, elija el Profile (morCar, morBike o morFoot) y llame a GetRoute. La ruta se devuelve en la misma clase TsgcMapRoute que usa TsgcMapDirections, así que se dibuja con sgcMapAddRouteToPolylines o se entrega a TsgcMapRouteCalculator. Grafos de rutas y enrutamiento sin conexión describe el grafo y la clase constructora.

Lazarus

sgcMaps tiene un paquete para Lazarus y Free Pascal: **Packages\Lazarus\sgcMapsLazarus.lpk**. Instala el control de mapa en la LCL, con el mismo modelo de mapa y el mismo código que el control VCL, pero con menos características, porque las partes que necesitan unidades de la RTL exclusivas de Delphi no se compilan bajo FPC. La diferencia más importante: **el control de Lazarus no puede descargar teselas, así que muestra únicamente teselas de una carpeta.**

Instalación del paquete

1. Inicie Lazarus y elija **Package**, **Open Package File (.lpk)**.
2. Abra **Packages\Lazarus\sgcMapsLazarus.lpk** de la instalación de sgcUI. El paquete encuentra las unidades de la carpeta Source mediante una ruta relativa, así que conserve la estructura de carpetas de la instalación.
3. Pulse **Compile**, después **Use**, **Install**, y deje que Lazarus reconstruya y reinicie el IDE.
4. TsgcVCLMap está ahora en la página **SGC Maps** de la paleta de componentes.

El paquete requiere los paquetes LCL, FCL e IDEIntf, que tiene toda instalación de Lazarus. Un proyecto que usa el mapa incluye sgcMapsLazarus en sus paquetes requeridos, algo que el IDE añade al colocar el control en un formulario.

Qué se incluye

Incluido	Unidad
El control de mapa en la LCL, TsgcVCLMap	sgcMaps_VCL
El modelo de mapa: capas, marcadores, polilíneas, polígonos, rectángulos, círculos, etiquetas, mapas de calor y opciones	sgcMaps_Classes
El origen de teselas de carpeta (archivos z/x/y)	sgcMaps_TileSource
La lista de proveedores y sgcMapExpandTileURL	sgcMaps_Providers
Las funciones de geometría	sgcMaps_Geometry
El rasterizador por software, TsgcMapRasterizer	sgcMaps_Raster

El control dibuja marcadores (con iconos PNG personalizados), agrupaciones, formas, etiquetas y mapas de calor, y guarda la vista con RenderToBitmap, SaveToBMP y SaveToPNG.

Qué no se incluye

- **Teselas de un servidor.** Una capa con un origen mtsHTTP o con un Provider no muestra teselas en Lazarus. Prepare una carpeta de teselas en Delphi con TsgcMapTileDownloader, o con cualquier herramienta que escriba teselas z/x/y, y distribúyala con la aplicación.
- **MBTiles y teselas vectoriales.** Estos orígenes necesitan FireDAC y System.ZLib.
- **GeoJSON, GPX y KML.** Las unidades figuran en el paquete pero se compilan vacías bajo FPC.
- **Los componentes no visuales.** Los servicios en línea, el calculador de rutas, el enrutador sin conexión, el geocodificador sin conexión, el lector de OSM y TsgcMapLocation están disponibles solo en Delphi.
- **Rotación del mapa.** Options.Bearing se ignora en la LCL.
- **Opacidad.** La opacidad de las capas, las polilíneas y los rellenos no se aplica en la LCL.

Ejemplo

El programa siguiente muestra un mapa a partir de una carpeta de teselas junto al ejecutable. Cree un proyecto Application nuevo, añada sgcMapsLazarus a sus paquetes requeridos y sustituya el programa principal por este código:

```
program LazarusMap;

{$mode delphi}{$H+}

uses
  Interfaces, Forms, Controls, SysUtils,
  sgcMaps_Classes, sgcMaps_VCL;

var
  oForm: TForm;
  oMap: TsgcVCLMap;
  oLayer: TsgcMapLayer;
begin
  Application.Initialize;
  Application.CreateForm(TForm, oForm);
  oForm.Caption := 'sgcMaps on Lazarus';
  oForm.SetBounds(100, 100, 900, 600);

  oMap := TsgcVCLMap.Create(oForm);
  oMap.Parent := oForm;
  oMap.Align := alClient;

  // the LCL control reads its tiles from a z\x\y folder
  oLayer := oMap.Layers.Add;
  oLayer.Name := 'Offline tiles';
  oLayer.TileSourceType := mtsFolder;
  oLayer.TileSourcePath := ExtractFilePath(ParamStr(0)) + 'tiles';
  oLayer.MaxZoom := 16;
  oLayer.Attribution := '(c) OpenStreetMap contributors';

  oMap.SetCenter(41.3874, 2.1686);
  oMap.Zoom := 12;
  oMap.AddMarker(41.3874, 2.1686, 'Barcelona');

  Application.Run;
end.
```

La carpeta **tiles** contiene una subcarpeta por nivel de zoom, una subcarpeta por columna y una imagen por tesela, como tiles\12\2072\1529.png. Establezca MaxZoom en el nivel de zoom más alto de la carpeta, para que el control no busque teselas que no existen. Conserve la atribución de las teselas, ya que las condiciones del proveedor de teselas se siguen aplicando a una copia en disco.

Véase también

- TsgcVCLMap
- Teselas sin conexión y el descargador de teselas
- Instalación

Información general de .NET

sgcMaps for .NET es un puerto gestionado en C# de la biblioteca Delphi sgcMaps, no un envoltorio sobre ella. Tiene el mismo modelo de mapa, los mismos proveedores de teselas, los mismos servicios en línea y el mismo enrutador y geocodificador sin conexión, y dibuja el mapa de forma nativa con SkiaSharp, sin control de navegador y sin JavaScript.
Los archivos son compatibles entre las dos bibliotecas: un grafo de rutas (.sgcgraph), un índice de GeoNames o una carpeta de teselas creados con Delphi se abren en .NET, y al revés. El conjunto de pruebas de QA de .NET lo comprueba con un grafo de rutas y un índice de GeoNames escritos por la biblioteca Delphi.

Paquetes

El puerto se divide en cuatro paquetes NuGet, así que un proyecto solo incorpora lo que usa. Los cuatro tienen la versión **2.0.0**: sgcMaps for .NET lleva sus propios números de versión, independientes de la versión de Delphi.

PackageId	Contiene	Frameworks de destino	Depende de
esegece.sgcMaps	El núcleo: el modelo de mapa (capas, marcadores, polilíneas, polígonos, rectángulos, círculos, etiquetas, mapas de calor y opciones), la matemática de teselas y la geometría, los proveedores de teselas, los orígenes de teselas de carpeta, MBTiles y HTTP, el descargador de teselas, la importación y exportación de GeoJSON, GPX y KML, los lectores de teselas vectoriales y de estilos, y los servicios en línea: Geocoder, Directions,	net8.0, net9.0	Microsoft.Data.Sqlite

	Places, Elevation, MapTimeZone y RouteCalculator.		
esegece.sgcMaps.Rendering	**MapView**, el motor de mapa. Dibuja teselas raster y vectoriales, superposiciones, agrupaciones, mapas de calor y ventanas emergentes con SkiaSharp, gestiona la entrada de puntero, rueda y teclado, y renderiza en una imagen o en un archivo PNG sin ventana.	net8.0, net9.0	esegece.sgcMaps, SkiaSharp, SkiaSharp.NativeAssets.Linux.NoDependencies
esegece.sgcMaps.WinForms	**SgcMapControl**, el control de mapa de Windows Forms, que aloja un MapView.	net8.0-windows, net9.0-windows	esegece.sgcMaps.Rendering
esegece.sgcMaps.Offline	Los servicios sin conexión: **OsmReader** (.osm.pbf y XML .osm), **RouteGraphBuilder** y **OfflineRouter**, **GeoIndexBuilder** y **OfflineGeocoder**.	net8.0, net9.0	esegece.sgcMaps

Solo edición Registered

sgcMaps for .NET es un producto Registered (comercial). No hay edición Community ni paquete **.Community**: los cuatro paquetes anteriores son los únicos.

Hosts

- **Windows Forms**: SgcMapControl, la contrapartida del TsgcVCLMap de Delphi.
- **Sin ventana**: MapView renderiza en un lienzo de SkiaSharp, una imagen o un archivo PNG, en Windows, Linux y macOS. Esto abarca aplicaciones de consola, servicios y servidores web.

- **Su propio host**: MapView no depende de ningún framework de interfaz de usuario. Dibuja en cualquier SKCanvas y recibe su entrada mediante PointerDown, PointerMove, PointerUp, Wheel, Pinch, Rotate y KeyDown, que es exactamente como lo maneja SgcMapControl.

No hay control WPF, Avalonia ni MAUI, ni contrapartida del control FireMonkey TsgcFMXMap.

Espacios de nombres

Espacio de nombres	Contenido
esegece.sgcMaps	El modelo de mapa, MapCoord, MapColors, TileMath, MapGeometry, MapProviders, los orígenes de teselas, TileDownloader, GeoJsonFormat, GpxFormat y KmlFormat.
esegece.sgcMaps.Services	Los servicios en línea (MapServiceBase, Geocoder, Directions, Places, Elevation, MapTimeZone), RouteCalculator y las clases de resultado, como GeocodingResult y Route.
esegece.sgcMaps.Vector	El lector de Mapbox Vector Tile y el lector de estilos de MapLibre: VectorTile, VectorStyle, StyleEvaluator.
esegece.sgcMaps.Rendering	MapView, VectorTileSource, VectorTileRenderer y las clases de argumentos de eventos.
esegece.sgcMaps.WinForms	SgcMapControl.
esegece.sgcMaps.Offline	OsmReader, RouteGraphBuilder, OfflineRouter, GeoIndexBuilder y OfflineGeocoder.

Las clases sin conexión comparten las clases de resultado de los servicios en línea: OfflineRouter rellena una **Route** y OfflineGeocoder devuelve objetos **GeocodingResult**, de modo que el código que muestra un resultado funciona con ambos. Por eso los ejemplos sin conexión también usan esegece.sgcMaps.Services.

Clases que hay que liberar

Estas clases contienen archivos, conexiones, subprocesos u objetos nativos de SkiaSharp, e implementan IDisposable. Libérelas con Dispose, o créelas con una instrucción using:

- **MapView**: sus cargadores de teselas y las imágenes de los iconos de los marcadores. **SgcMapControl** libera su propio MapView, su búfer trasero y su temporizador.
- **Geocoder**, **Directions**, **Places**, **Elevation** y **MapTimeZone**: un HttpClient propio, cuando se crearon con un HttpMessageHandler.
- **TileDownloader**: su HttpClient.
- **OfflineRouter** y **OfflineGeocoder**: Dispose llama a Close, que libera el grafo de rutas o la conexión SQLite.

Véase también

- Inicio Rápido: WinForms
- Renderizado sin interfaz visual
- Descripción general (Delphi)
- Instalación de .NET
- Correspondencia de clases
- Diferencias de .NET respecto a Delphi

Instalación de .NET

El instalador de sgcUI instala los paquetes de Delphi, C++Builder y Lazarus. sgcMaps for .NET se distribuye, en cambio, como sus propios paquetes NuGet, solo para la edición Registered.

Adición de los paquetes

Añada el paquete del host que utilice. Cada paquete trae consigo aquellos de los que depende, así que una aplicación Windows Forms solo necesita esegece.sgcMaps.WinForms, y un renderizador sin interfaz visual solo necesita esegece.sgcMaps.Rendering.

```
dotnet add package esegece.sgcMaps.WinForms --version 2.0.0
dotnet add package esegece.sgcMaps.Rendering --version 2.0.0
dotnet add package esegece.sgcMaps.Offline --version 2.0.0
dotnet add package esegece.sgcMaps --version 2.0.0
```

Paquete	Añádalo para	Frameworks de destino
esegece.sgcMaps.WinForms	Un control de mapa en un formulario de Windows Forms.	net8.0-windows, net9.0-windows
esegece.sgcMaps.Rendering	Mapas renderizados sin ventana, o un mapa en un host propio.	net8.0, net9.0
esegece.sgcMaps.Offline	Enrutamiento sin conexión, geocodificación sin conexión y lectura de archivos de OpenStreetMap.	net8.0, net9.0
esegece.sgcMaps	Solo el modelo de mapa, los formatos, el descargador de teselas y los servicios en línea, sin dibujar nada.	net8.0, net9.0

Las mismas líneas en un archivo de proyecto:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcMaps.WinForms" Version="2.0.0" />
  <PackageReference Include="esegece.sgcMaps.Offline" Version="2.0.0" />
</ItemGroup>
```

Cuando los paquetes vienen como archivos .nupkg en lugar de un feed, añada primero su carpeta como origen de paquetes:

```
dotnet nuget add source C:\Packages\sgcMaps --name sgcMaps
```

Requisitos

- .NET 8 o .NET 9. Los paquetes no se dirigen a .NET Framework ni a .NET Standard.
- El control Windows Forms necesita Windows. Los otros tres paquetes funcionan en Windows, Linux y macOS.
- esegece.sgcMaps usa **Microsoft.Data.Sqlite** para leer archivos MBTiles y el índice de GeoNames. No hay que instalar nada para ello.
- esegece.sgcMaps.Rendering usa **SkiaSharp** y trae consigo **SkiaSharp.NativeAssets.Linux.NoDependencies**, de modo que también renderiza en un servidor Linux o en un contenedor sin ninguna biblioteca nativa adicional.

Sin edición Community

No hay edición Community (gratuita) de sgcMaps for .NET, ni paquete con el sufijo **.Community**. Todos los paquetes anteriores necesitan una licencia Registered.

Véase también

- Inicio Rápido: WinForms
- Instalación (Delphi)
- Información general de .NET
- Correspondencia de clases
- Diferencias de .NET respecto a Delphi

Inicio Rápido: WinForms

SgcMapControl es el control de mapa de Windows Forms, la contrapartida en .NET del TsgcVCLMap de Delphi. Se distribuye en el paquete **esegece.sgcMaps.WinForms**, para net8.0-windows y net9.0-windows, y trae consigo esegece.sgcMaps.Rendering y esegece.sgcMaps.
El control aloja un MapView, el motor de mapa, en su propiedad **Map**. Pinta la vista con SkiaSharp en un búfer trasero y le reenvía el ratón, la rueda y el teclado. Las teselas se cargan en subprocesos en segundo plano, de modo que la interfaz de usuario nunca espera a la red; el control se repinta en el subproceso de la interfaz cuando llegan, y también dispara TileError en el subproceso de la interfaz.

1. Crear el proyecto

```
dotnet new winforms -n MapQuickStart
cd MapQuickStart
dotnet add package esegece.sgcMaps.WinForms --version 2.0.0
```

2. Mostrar un mapa

Sustituya Program.cs por el programa siguiente y elimine Form1.cs y Form1.Designer.cs. Añade una capa de OpenStreetMap con caché en disco, centra el mapa en Madrid y añade un marcador arrastrable y una línea.
MapProviders.ApplyProvider copia la configuración del proveedor en la capa: TileSourceType, UrlTemplate, MaxZoom y Attribution. Una capa cuyo Provider no es Custom descarga las teselas de su proveedor incluso sin la llamada, así que la llamada solo hace falta cuando se quiere leer o cambiar esos valores.

3. Ejecutarlo

```
dotnet run
```

Arrastre el mapa para desplazarlo, gire la rueda para acercar o alejar y arrastre el marcador. El título del formulario muestra el último clic, la posición en la que se soltó el marcador o la tesela que falló.

Propiedades

Propiedad	Tipo	Descripción
Map	MapView	El motor de mapa. Métodos como AddMarker, AddPolyline, FitToMarkers, SetLayerVisible y RenderToPng se llaman sobre él.
CenterLatitude, CenterLongitude	double	El centro de la vista, en grados.
Zoom	int	El nivel de zoom, limitado por Options.MinZoom y Options.MaxZoom.
Bearing	double	La rotación del mapa en grados en sentido horario desde el norte. El usuario puede girar el mapa cuando Options.RotationEnabled es true.
Options	MapOptions	El comportamiento y las decoraciones de la vista: límites de zoom, TileSize, botones de zoom, barra de escala, atribución, color de fondo, arrastre, zoom, rotación y el selector de capas.

MarkerOptions	MarkerOptions	El tamaño, el color y la fuente predeterminados de los marcadores, y las opciones de agrupación en MarkerOptions.Clustering.
Layers	MapItemCollection<Layer>	Las capas de teselas.
Markers, Polylines, Polygons, Circles, Rectangles, Labels, Heatmaps	MapItemCollection<T>	Las superposiciones. Añada un elemento nuevo con Add(), o con los métodos Add de Map.

Las propiedades habituales de Control, como Dock, Anchor, Enabled, Visible y TabStop, también están disponibles.

Eventos

Evento	Argumentos	Se produce cuando
MapClick	MapClickEventArgs: Latitude, Longitude	Se hace clic en el mapa.
MarkerClick	MapItemEventArgs<Marker>: Item	Se hace clic en un marcador.
MarkerDragStart, MarkerDrag, MarkerDragEnd	MapItemEventArgs<Marker>: Item	Se arrastra un marcador con Draggable establecido en true.
ShapeEdited	MapItemEventArgs<MapItem>: Item	El usuario ha movido, añadido o eliminado un vértice de una polilínea o un polígono editable.
ClusterClick	ClusterClickEventArgs: Count, Latitude, Longitude, Handled	Se hace clic en una agrupación de marcadores. Establezca Handled en true para gestionar el clic usted mismo.
LayerChanged	MapItemEventArgs<Layer>: Item	Se muestra u oculta una capa, por ejemplo desde el selector de capas.
ZoomChanged, BearingChanged	EventArgs	Cambia el nivel de zoom o la rotación.
TileError	TileErrorEventArgs: Layer, Zoom, X, Y, Error	No se puede cargar una tesela, por ejemplo con un error HTTP del servidor de teselas.

Son eventos normales de C#: se pueden asociar varios manejadores con +=.

Véase también

- Renderizado sin interfaz visual
- Demos de .NET
- Teselas y capas
- Información general de .NET
- Instalación de .NET
- Correspondencia de clases
- Diferencias de .NET respecto a Delphi

Renderizado sin interfaz visual

MapView, en el paquete **esegece.sgcMaps.Rendering**, es el motor de mapa de sgcMaps for .NET. SgcMapControl lo muestra en una ventana, pero no necesita ninguna ventana: renderiza en un lienzo de SkiaSharp, una imagen o un archivo PNG. Una aplicación de consola, un servicio de Windows, un demonio de Linux o un servidor web puede por tanto producir mapas estáticos, con las mismas capas y superposiciones que el control. Es la contrapartida en .NET del uso sin interfaz visual de los controles de Delphi descrito en Renderizado de mapas estáticos.

Renderizado a un archivo PNG

El programa siguiente renderiza la Torre Eiffel con zoom 15 con un marcador y un círculo, y escribe un archivo PNG de 1024 x 768.
RenderToPng dibuja la vista, espera las teselas que aún se están cargando y la dibuja de nuevo hasta que han llegado todas las teselas o ha pasado **tileTimeout** (30 segundos cuando no se indica). Tras la llamada, **LastRenderMissingTiles** indica cuántas teselas faltaban en la imagen. El tamaño de la vista se restaura después, de modo que un MapView puede renderizar imágenes de varios tamaños.

Renderizado en memoria

RenderToImage devuelve un SKImage en lugar de escribir un archivo. Codifíquelo en el formato que necesite, por ejemplo para responder a una petición web con bytes PNG:

Dibujo en su propio lienzo

Render(SKCanvas canvas, int width, int height, bool includeControls) dibuja la vista una vez en cualquier lienzo de SkiaSharp y no espera. Las teselas que aún no están cargadas se ponen en cola en segundo plano, y el evento **Invalidated** solicita el siguiente pintado cuando llegan. Invalidated puede dispararse en un subproceso de trabajo, así que un host debe cambiar a su propio subproceso de interfaz antes de pintar, como hace SgcMapControl.

Qué se dibuja

- Las capas de teselas, las etiquetas de las teselas vectoriales, los mapas de calor, polígonos, rectángulos, polilíneas, círculos, marcadores y etiquetas, en ese orden.
- Los botones de zoom, el selector de capas y la brújula solo cuando **includeControls** es true. Es false de forma predeterminada en RenderToImage y RenderToPng.
- La barra de escala y la atribución cuando **Options.ShowScaleBar** y **Options.ShowAttribution** son true, diga lo que diga includeControls. Mantenga la atribución visible: los proveedores de teselas la exigen.

Miembros para el uso sin interfaz visual

Miembro	Descripción
SetSize(int width, int height)	Establece el tamaño de la vista, que usan FitBounds, FitToMarkers y las conversiones de coordenadas antes de renderizar.
SetCenter(double latitude, double longitude), Zoom, Bearing	La vista.
FitBounds, FitToMarkers(int paddingPx), FitToPolyline(Polyline polyline, int paddingPx)	Eligen el centro y el zoom que muestran un área, los marcadores o una polilínea.

AddMarker, AddPolyline, AddPolygon, AddCircle, AddRectangle, AddLabel	Añaden superposiciones. AddMarker recibe el texto de la ventana emergente y de la sugerencia.
Render, RenderToImage, RenderToPng	Dibujan la vista, como se describe arriba.
WaitForTiles(TimeSpan timeout)	Bloquea hasta que las teselas en cola se han cargado o vence el tiempo de espera, y devuelve true cuando no queda nada pendiente.
PendingTileCount	El número de teselas que se están cargando en segundo plano.
LastRenderMissingTiles	El número de teselas que faltaban en el último renderizado.
DrawnMarkerCount, DrawnClusterCount, DrawnLabelCount	Lo que dibujó el último renderizado.
AttributionText()	La atribución de las capas visibles, para imprimirla junto a la imagen cuando no está dibujada en ella.
LatLonToScreen, ScreenToLatLon	Convierten entre coordenadas y píxeles de la vista.

Linux y macOS

El paquete Rendering se dirige a net8.0 y net9.0 sin dependencia de Windows, y trae consigo la biblioteca nativa de SkiaSharp para Linux (SkiaSharp.NativeAssets.Linux.NoDependencies), que no necesita fontconfig ni otro paquete del sistema. Un renderizador sin interfaz visual puede por tanto ejecutarse en un contenedor Linux. Las teselas de un servidor necesitan acceso HTTPS de salida; las teselas de una carpeta o de un archivo MBTiles no necesitan ninguno.

Véase también

- Demos de .NET
- Datos sin conexión de .NET
- 05 Renderizado estático (Delphi)
- Información general de .NET
- Instalación de .NET
- Correspondencia de clases
- Diferencias de .NET respecto a Delphi

Datos sin conexión de .NET

sgcMaps for .NET puede mostrar un mapa, calcular rutas y geocodificar sin conexión de red. Usa tres tipos de datos locales: teselas en disco, un grafo de rutas creado a partir de OpenStreetMap y un índice de GeoNames. Todos son compatibles a nivel de archivo con la biblioteca Delphi, de modo que los datos creados con las herramientas de Delphi se pueden usar aquí, y al revés.

Datos	Creado con	Usado por	Paquete
Carpeta de teselas (z/x/y)	TileDownloader, o cualquier herramienta de teselas	Una capa con TileSourceType.Folder	esegece.sgcMaps
Archivo MBTiles, raster o vectorial	Cualquier herramienta de MBTiles	Una capa con TileSourceType.MBTiles o TileSourceType.VectorMBTiles	esegece.sgcMaps, esegece.sgcMaps.Rendering para teselas vectoriales
Grafo de rutas (.sgcgraph)	RouteGraphBuilder	OfflineRouter	esegece.sgcMaps.Offline
Índice de GeoNames (SQLite)	GeoIndexBuilder	OfflineGeocoder	esegece.sgcMaps.Offline

Teselas sin conexión

Una capa lee sus teselas de una carpeta o de un archivo MBTiles cuando **TileSourceType** es Folder o MBTiles y **TileSourcePath** apunta a él. Los archivos MBTiles vectoriales se dibujan con el estilo de MapLibre indicado en **VectorStyleFile**, o con el estilo integrado cuando está vacío. Consulte Teselas sin conexión y Teselas vectoriales.

Descarga de un paquete de teselas

TileDownloader descarga las teselas de un área y de un rango de niveles de zoom en una carpeta z/x/y, y omite las teselas que ya están allí. No tiene un UrlTemplate predeterminado a propósito: tile.openstreetmap.org y la mayoría de los servidores públicos de teselas prohíben las descargas masivas, así que apúntelo a su propio servidor o a un proveedor cuyas condiciones las permitan, y mantenga DelayMS por encima de cero.

Enrutamiento sin conexión

RouteGraphBuilder lee un extracto de OpenStreetMap, un archivo .osm.pbf o un archivo XML .osm, y escribe la red viaria como un archivo .sgcgraph con un perfil de coche, de bicicleta y a pie. **OfflineRouter** carga ese archivo y responde a rutas sin red. La ruta es una **Route**, la misma clase que devuelve el servicio Directions en línea, con la distancia en metros, la duración en segundos, los puntos, los tramos y los pasos.

Miembro	Descripción
RouteGraphBuilder.Build(string inputFile, string outputFile)	Construye el grafo. Sus estadísticas están en WayCount, NodeCount, MissingNodes, VertexCount, EdgeCount, ShapeCount y NameCount, y Progress informa de los bytes leídos.

OfflineRouter.GraphFile, Open(), Close(), Active	El archivo del grafo, y su apertura o cierre. Active es de solo lectura; el constructor que recibe un nombre de archivo solo establece GraphFile.
OfflineRouter.Profile	OfflineRouterProfile.Car (predeterminado), Bike o Foot.
OfflineRouter.GetRoute(Route route, params MapCoord[] waypoints)	Calcula la ruta a través de dos o más puntos de paso. Devuelve false y establece LastError cuando no hay ruta. Una sobrecarga recibe un IReadOnlyList<MapCoord>.

Construir el grafo de un país entero requiere tiempo y memoria; constrúyalo una vez y distribuya el archivo .sgcgraph. Consulte Enrutamiento sin conexión para conocer las fases de la construcción.

Geocodificación sin conexión

GeoIndexBuilder convierte un volcado de GeoNames en un índice SQLite, y **OfflineGeocoder** lo busca por nombre o por posición. Descargue un volcado como cities1000.txt, junto con admin1CodesASCII.txt y countryInfo.txt, del servidor de exportación de GeoNames. Los resultados son objetos **GeocodingResult**, igual que con el Geocoder en línea.

Miembro	Valor predeterminado	Descripción
GeoIndexBuilder.InputFile, DatabaseFile		El volcado de GeoNames que se lee y el índice que se escribe.
GeoIndexBuilder.Admin1CodesFile, CountryInfoFile		Los archivos que convierten códigos en nombres de regiones y países.
GeoIndexBuilder.FeatureClasses	PA	Las clases de característica de GeoNames que se importan: P para lugares habitados, A para áreas administrativas.
GeoIndexBuilder.MinPopulation	0	Omite los lugares con una población menor.
GeoIndexBuilder.IncludeAlternateNames	true	Indexa también los nombres alternativos, como los nombres en otros idiomas.
GeoIndexBuilder.BatchSize	2000	Filas escritas por lote.
OfflineGeocoder.MaxResults	5	El número máximo de resultados.
OfflineGeocoder.CountryCodes		Códigos de país ISO alfa-2 separados por comas, como ES,PT, que restringen cada búsqueda.
OfflineGeocoder.Geocode, ReverseGeocode, ReverseGeocodeCountry		Buscan por nombre, buscan los lugares más cercanos a una posición y devuelven el país de una posición.

OfflineGeocoder, igual que OfflineRouter, se abre con Open() y se cierra con Close() o Dispose().

Lectura de archivos de OpenStreetMap

OsmReader lee en flujo un archivo .osm.pbf o XML .osm y dispara **Node**, **Way** y **Relation** para cada elemento, con sus etiquetas, y **Progress** con los bytes leídos. ReadNodes, ReadWays, ReadRelations, SkipNodeTags y WayFilter omiten lo que no necesita. RouteGraphBuilder está construido sobre él, y sirve para cualquier otra importación de datos de OpenStreetMap.

Véase también

- Demos de .NET
- Geocodificación sin conexión
- Información general de .NET
- Instalación de .NET
- Correspondencia de clases
- Diferencias de .NET respecto a Delphi

Correspondencia de clases

La tabla asigna cada componente y clase Delphi de sgcMaps a su contrapartida en .NET. Las clases .NET omiten el prefijo Tsgc y usan los nombres habituales de .NET; sus miembros conservan los nombres de Delphi siempre que el concepto es el mismo.

Controles de mapa y modelo

Delphi	.NET	Paquete
TsgcVCLMap	SgcMapControl (Windows Forms), con su motor en SgcMapControl.Map	esegece.sgcMaps.WinForms
TsgcFMXMap	Sin contrapartida. MapView puede alojarse en cualquier framework que dibuje con SkiaSharp.	esegece.sgcMaps.Rendering
Un control de mapa que renderiza sin ventana, TsgcMapRasterizer	MapView: Render, RenderToImage, RenderToPng	esegece.sgcMaps.Rendering
TsgcMapLayer, TsgcMapMarker, TsgcMapPolyline, TsgcMapPolygon, TsgcMapRectangle, TsgcMapCircle, TsgcMapLabel, TsgcMapHeatmap	Layer, Marker, Polyline, Polygon (agujeros en PolygonHoles), Rectangle, Circle, Label, Heatmap	esegece.sgcMaps
TsgcMapLayers, TsgcMapMarkers y las demás colecciones	MapItemCollection<T>	esegece.sgcMaps
TsgcMapOptions, TsgcMapMarkerOptions, TsgcMapClusterOptions	MapOptions, MarkerOptions, ClusterOptions	esegece.sgcMaps
TsgcMapCoord, TsgcMapHeatmapPoint	MapCoord, HeatmapPoint (readonly record structs)	esegece.sgcMaps

Teselas, proveedores y formatos

Delphi	.NET	Paquete
TsgcMapTileProvider, TsgcMapTileSourceType	TileProvider, TileSourceType	esegece.sgcMaps
sgcMapApplyProvider, sgcMapGetProviderInfo, sgcMapProviderNames, sgcMapExpandTileURL, TsgcMapProviderInfo	MapProviders.ApplyProvider, GetInfo, ProviderNames, ExpandTileUrl, ProviderInfo	esegece.sgcMaps
Los orígenes de teselas de carpeta, MBTiles y HTTP	FolderTileSource, MBTilesTileSource, HttpTileSource (ITileSource), TileSourceFactory	esegece.sgcMaps
TsgcMapTileSource_MVT	VectorTileSource	esegece.sgcMaps.Rendering

TsgcMapVectorTile, TsgcMapVectorLayer, TsgcMapVectorFeature, TsgcMapVectorStyle	VectorTile, VectorLayer, VectorFeature, VectorStyle (namespace esegece.sgcMaps.Vector)	esegece.sgcMaps
TsgcMapTileDownloader	TileDownloader	esegece.sgcMaps
TsgcMapGeoJSON, TsgcMapGPX, TsgcMapKML	GeoJsonFormat, GpxFormat, KmlFormat	esegece.sgcMaps
TsgcMapGPXTrackStats, TsgcMapElevationGradient	GpxTrackStats, ElevationGradient	esegece.sgcMaps
Las funciones de sgcMaps_Geometry (sgcMapDistance, sgcMapInitialBearing, sgcMapDestinationPoint, sgcMapGeodesicInterpolate, sgcMapPointInPolygon, sgcMapSimplify, sgcMapNearestPointOnPolyline, sgcMapClusterPoints, sgcMapRenderHeatmap)	La clase estática MapGeometry: Distance, InitialBearing, DestinationPoint, GeodesicInterpolate, PointInPolygon, Simplify, NearestPointOnPolyline, ClusterPoints, RenderHeatmap	esegece.sgcMaps
La matemática de teselas de sgcMaps_Classes (sgcMapLatLonToTileF, sgcMapTileFToLatLon, sgcMapLatLonToPixel, sgcMapPixelToLatLon, sgcMapTileCount)	La clase estática TileMath: TryLatLonToTile, TryTileToLatLon, TryLatLonToPixel, TryPixelToLatLon, TileCount	esegece.sgcMaps

Servicios en línea y enrutamiento

Delphi	.NET	Paquete
TsgcMapService	MapServiceBase	esegece.sgcMaps
TsgcMapGeocoder	Geocoder, con GeocodingResult y GeocodingProvider	esegece.sgcMaps
TsgcMapDirections	Directions, con Route, RouteLeg, RouteStep, DirectionsProvider, TravelMode y PolylineCodec	esegece.sgcMaps
sgcMapAddRouteToPolylines	Route.AddToPolylines	esegece.sgcMaps
TsgcMapPlaces	Places, con Place y PlacesProvider	esegece.sgcMaps
TsgcMapElevation	Elevation, con ElevationResult y ElevationProvider	esegece.sgcMaps
TsgcMapTimeZone	MapTimeZone, con MapTimeZoneInfo y TimeZoneProvider	esegece.sgcMaps
TsgcMapRouteCalculator	RouteCalculator	esegece.sgcMaps

Offline

Delphi	.NET	Paquete
TsgcMapOfflineRouter	OfflineRouter, con OfflineRouterProfile	esegece.sgcMaps.Offline

TsgcMapRouteGraphBuilder, la herramienta RouteGraphBuilder	RouteGraphBuilder	esegece.sgcMaps.Offline
TsgcMapOfflineGeocoder	OfflineGeocoder	esegece.sgcMaps.Offline
TsgcMapGeoIndexBuilder, la herramienta GeoIndexBuilder	GeoIndexBuilder	esegece.sgcMaps.Offline
TsgcMapOSMReader, TsgcMapOSMTags, TsgcMapOSMMember	OsmReader, OsmTags, OsmMember	esegece.sgcMaps.Offline

Sin contrapartida en .NET

- TsgcMapLocation, la posición del dispositivo.
- Los controles FireMonkey y Lazarus.
- Las herramientas de línea de comandos GeoIndexBuilder y RouteGraphBuilder. Llame a las clases del mismo nombre, o use OfflineDemo, que crea los dos archivos.

Véase también

- Información general de .NET
- Instalación de .NET
- Diferencias de .NET respecto a Delphi

Diferencias de .NET respecto a Delphi

El puerto .NET conserva el modelo, el comportamiento y los formatos de archivo de la biblioteca Delphi. Lo que cambia es la forma de la API, donde .NET tiene sus propias convenciones.

Nombres

- Las clases omiten el prefijo Tsgc: TsgcMapGeocoder pasa a ser Geocoder y TsgcVCLMap pasa a ser SgcMapControl. La Correspondencia de clases las enumera todas.
- Las enumeraciones también omiten sus prefijos: mtpOpenStreetMap pasa a ser TileProvider.OpenStreetMap, y mtsMBTiles pasa a ser TileSourceType.MBTiles.
- Algunos miembros siguen la capitalización de .NET, como UrlTemplate y NominatimUrl. La mayoría conserva sus nombres de Delphi.

Un motor, un control

En Delphi los controles de mapa contienen el mapa. En .NET el mapa es un **MapView**, que funciona sin ventana, y **SgcMapControl** es un control de Windows Forms que aloja uno en su propiedad Map. El control expone directamente la vista y las colecciones (Zoom, Layers, Markers...), y métodos como AddMarker, FitToMarkers o RenderToPng se llaman sobre SgcMapControl.Map.

Los servicios en línea son asíncronos

En Delphi, cada petición de un servicio tiene un método síncrono y un método Async que responde en un evento de resultado, como OnGeocodeResult. En .NET cada petición es un único método basado en Task que devuelve el resultado: GeocodeAsync y ReverseGeocodeAsync, GetRouteAsync, SuggestAsync y RetrieveAsync, GetElevationsAsync y GetElevationAlongPathAsync, GetTimeZoneAsync, y RouteCalculator.CalculateAsync. Todos reciben un CancellationToken opcional. No hay eventos de resultado, y no se necesita CheckSynchronize en una aplicación de consola.

- Una petición fallida lanza una **ServiceException**, con el estado HTTP en StatusCode (0 cuando no hubo respuesta HTTP). También establece LastError y dispara el evento Error, como OnError en Delphi.
- Una petición sin resultado no es un error: la lista queda vacía, y Geocoder establece LastError en "No results".
- Los servicios comparten un único HttpClient de forma predeterminada. Un constructor que recibe un HttpMessageHandler o un HttpClient permite usar un proxy o un manejador de pruebas.
- El espaciado, en todo el proceso, de las peticiones a Nominatim, al servidor de demostración de OSRM y a OpenTopoData, de una por segundo, es el mismo que en Delphi.

Eventos

Los eventos son eventos de C# de tipo EventHandler<T>, como MapClick con MapClickEventArgs. Se pueden asociar varios manejadores con += y quitarlos con -=, mientras que un evento de Delphi contiene un solo método. Los datos de un parámetro de evento de Delphi son una propiedad de los argumentos: e.Item para un marcador, e.Latitude y e.Longitude para un clic, e.Handled para detener la acción predeterminada.

Colores

Los colores son valores uint en orden ARGB, 0xAARRGGBB, como MapColors.Blue (0xFF0000FF), donde el modelo de Delphi usa TColor. **MapColors.FromTColor** y **MapColors.ToTColor** convierten entre ambos, y **MapColors.FromRgb** crea un color opaco.

Coordenadas y colecciones

- MapCoord y HeatmapPoint son readonly record structs, que se crean con new MapCoord(latitude, longitude).
- Las colecciones de superposiciones son MapItemCollection<T>, una Collection<T> con Add(), BeginUpdate, EndUpdate y un evento Changed.
- El icono de un marcador es el contenido de un archivo de imagen en Marker.Icon, una matriz de bytes, que se carga con LoadIconFromFile o LoadIconFromStream.

Enrutador y geocodificador sin conexión

OfflineRouter y OfflineGeocoder se abren con **Open()** y se cierran con **Close()** o Dispose(). Su propiedad Active es de solo lectura, mientras que en Delphi establecer Active en True los abre. OfflineRouter.GetRoute y OfflineGeocoder.Geocode son síncronos, como en Delphi.

Renderizado

El motor dibuja con SkiaSharp en todas las plataformas. RenderToPng escribe archivos PNG; para otro formato, codifique el SKImage de RenderToImage. Consulte Renderizado sin interfaz visual.

Plataformas y versiones

- Los paquetes se dirigen a .NET 8 y .NET 9, y el control se dirige a net8.0-windows y net9.0-windows. No hay compilación para .NET Framework.
- La versión de los paquetes es 2.0.0, independiente de la versión de Delphi.
- No hay edición Community. Consulte Instalación de .NET.
- TsgcMapLocation, el control FireMonkey y las herramientas de línea de comandos no tienen contrapartida.

Véase también

- Información general de .NET
- Instalación de .NET
- Correspondencia de clases

Demos de .NET

Con sgcMaps for .NET vienen tres demos, en la carpeta **.net\demos\Maps** de sgcUI. Cada una es un pequeño proyecto de estilo SDK para net8.0 o net8.0-windows. Referencia los proyectos de sgcMaps cuando el código fuente de la biblioteca está al lado (.net\source\Maps), y los paquetes NuGet de la versión 2.0.0 en caso contrario.

Demo	Tipo	Paquetes	Muestra
Console-RenderDemo	Console, net8.0	esegece.sgcMaps, esegece.sgcMaps.Rendering	Un mapa renderizado en un archivo PNG sin ventana, con marcadores y una ruta del servicio Directions en línea.
OfflineDemo	Console, net8.0	esegece.sgcMaps, esegece.sgcMaps.Offline	Enrutamiento y geocodificación sin conexión, incluida la construcción del grafo de rutas y del índice de GeoNames.
WinForms-Demo	Windows Forms, net8.0-windows	esegece.sgcMaps.WinForms	El control de mapa con dos capas base, marcadores, formas editables, un círculo, un mapa de calor y los eventos.

ConsoleRenderDemo

La versión .NET de la demo de Delphi 05 Renderizado estático. Sin argumentos, renderiza Madrid con zoom 13 con la ruta en coche desde la Puerta del Sol hasta el Parque del Retiro, obtenida del servidor público de OSRM, y escribe **madrid.png** en la carpeta actual. Las teselas se guardan en caché en una carpeta tile_cache junto al ejecutable.

```
dotnet run --project ConsoleRenderDemo
dotnet run --project ConsoleRenderDemo -- --center 48.8584,2.2945 --zoom 15 --size 800x600 --provider carto --marker 48.858
```

Argumento	Descripción
--center lat,lon	El centro del mapa.
--zoom z	El nivel de zoom, 13 de forma predeterminada.
--size WxH	El tamaño de la imagen, 1024x768 de forma predeterminada.
--provider osm|opentopo|esri|carto	El proveedor de teselas, osm de forma predeterminada.
--marker lat,lon	Añade un marcador. Se puede repetir.
--route lat,lon;lat,lon[;...]	Pide al servidor de OSRM una ruta a través de los puntos y la dibuja.
--out file.png	El archivo de salida, madrid.png de forma predeterminada.

La demo imprime el proveedor, la vista, la ruta, la ruta completa del archivo y el texto de atribución. Necesita acceso a internet para las teselas y la ruta.

OfflineDemo

Calcula rutas y geocodifica sin ninguna llamada de red. Construye el grafo de rutas a partir de un extracto de OpenStreetMap, o reutiliza el archivo .sgcgraph cuando existe, y calcula la ruta entre dos puntos en coche. Cuan-

do se indican los argumentos de GeoNames, también construye el índice de GeoNames, o lo reutiliza, y geocodifica una consulta.

```
dotnet run --project OfflineDemo -- --pbf andorra-latest.osm.pbf --from 42.5063,1.5218 --to 42.5350,1.5807
dotnet run --project OfflineDemo -- --pbf andorra-latest.osm.pbf --from 42.5063,1.5218 --to 42.5350,1.5807 --geonames citie
```

Argumento	Descripción
--pbf file.osm.pbf	El extracto de OpenStreetMap. Obligatorio.
--from lat,lon, --to lat,lon	El inicio y el final de la ruta. Obligatorios.
--graph file.sgcgraph	El grafo de rutas. De forma predeterminada, el nombre del extracto con la extensión .sgcgraph.
--geonames, --admin1, --countryinfo	El volcado de GeoNames, admin1CodesASCII.txt y countryInfo.txt.
--geocode "query"	El lugar que se busca.
--db file.sqlite	El índice de GeoNames. De forma predeterminada, el nombre del volcado con la extensión .sqlite.

Los extractos están disponibles en los servidores de descarga de OpenStreetMap, como Geofabrik, y los archivos de GeoNames en el servidor de exportación de GeoNames. Consulte Datos sin conexión de .NET.

WinFormsDemo

Un formulario con un **SgcMapControl** y una barra de estado. Muestra:

- Dos capas base, OpenStreetMap y OpenTopoMap, con el selector de capas y la rotación habilitada.
- Marcadores arrastrables en Madrid y Barcelona, y un marcador en Valencia.
- Una polilínea editable de Madrid a Barcelona pasando por Zaragoza, y un polígono editable.
- Un círculo de 40 km alrededor de Valencia, y un mapa de calor de 200 puntos ponderados alrededor de Sevilla.
- Los eventos MarkerClick, MapClick, MarkerDragEnd, ShapeEdited, ZoomChanged y TileError, escritos en la barra de estado.

Cuando se muestra el formulario, el mapa se ajusta a los marcadores. Las teselas se guardan en caché en la carpeta sgcMaps_tiles de la carpeta temporal.

```
dotnet run --project WinFormsDemo
```

Véase también

- Inicio Rápido: WinForms
- Renderizado sin interfaz visual
- Demos (Delphi)
- Información general de .NET
- Instalación de .NET
- Correspondencia de clases
- Diferencias de .NET respecto a Delphi

Solución de problemas

Los problemas siguientes son los que se encuentran con más frecuencia con sgcMaps, con su causa y su solución.

El mapa no muestra teselas

- Gestione **OnTileError**. Se dispara una vez por cada tesela que falla, con la capa, las coordenadas de la tesela y el error, como HTTP 403 Forbidden. Una tesela que el servidor no tiene (HTTP 404) no es un error: solo se cuenta en **LastRenderMissingTiles**, igual que una tesela que falta en una carpeta.
- Cuando una capa no se puede abrir en absoluto, el mapa dibuja el motivo en la vista, después del texto "Tile source not available". El más habitual es "Tile source directory not found": la capa no tiene Provider y su TileSourceType sigue siendo mtsFolder, el valor predeterminado, con un TileSourcePath que no existe.
- Las teselas de un servidor necesitan Delphi 2009 o posterior, y el control de Lazarus muestra únicamente teselas de carpeta (véase Lazarus). MBTiles y MBTiles vectorial necesitan Delphi XE7; en versiones anteriores, la capa informa de que el tipo de origen de teselas no está disponible en este compilador.
- Desde Delphi 2009 hasta XE7 las teselas se descargan con Indy, y https necesita las bibliotecas OpenSSL 1.0.2 (libeay32.dll y ssleay32.dll) junto al ejecutable. Sin ellas, el error indica que no se pudieron cargar las bibliotecas OpenSSL. A partir de Delphi XE8 el mapa usa THTTPClient y no necesita ninguna DLL.
- Una tesela fallida no se almacena, así que el mapa la vuelve a pedir en el siguiente repintado. OnTileError no se dispara dos veces para la misma tesela hasta que se vuelven a abrir las capas.

```
procedure TForm3.sgcVCLMap1TileError(Sender: TObject; aLayer: TsgcMapLayer;
  aZoom, aX, aY: Integer; const aError: string);
begin
  // raised once per tile; a tile the server does not have (404) is no error
  memoLog.Lines.Add(Format('%s %d/%d/%d: %s', [aLayer.Name, aZoom, aX, aY,
    aError]));
end;

procedure TForm3.btnMissingClick(Sender: TObject);
begin
  // tiles that were not available yet in the last paint
  memoLog.Lines.Add(Format('Missing tiles: %d',
    [sgcVCLMap1.LastRenderMissingTiles]));
end;
```

Claves de API

- Los proveedores de teselas MapTiler, Thunderforest y Stadia necesitan una clave de API en la propiedad **ApiKey** de la capa. Sin clave, OnTileError informa "This tile provider requires an API key (ApiKey)"; con una clave incorrecta, el proveedor responde HTTP 401 o 403.
- Los servicios en línea no comprueban la clave antes de enviar una solicitud. El error del proveedor llega en **LastError** (métodos síncronos) o en **OnError** (métodos asíncronos), como HTTP 403 con la explicación del proveedor.
- TsgcMapTimeZone con GeoNames toma en ApiKey su nombre de usuario de GeoNames, no una clave.
- No distribuya nunca una clave en una demo ni en código fuente público. Las demos de sgcMaps tienen en su lugar un cuadro de edición para la clave.

Políticas de uso y agente de usuario

Los servidores públicos gratuitos los mantienen voluntarios o patrocinadores, y cada uno tiene una política de uso. Una aplicación que la incumple puede ser bloqueada sin previo aviso.

- **tile.openstreetmap.org** permite únicamente la visualización interactiva: sin descargas masivas, sin precarga de áreas, con un agente de usuario que identifique a la aplicación y con una atribución visible. Las solicitudes de teselas de los controles de mapa se identifican como sgcMaps/1.0. Para una aplicación en producción use un proveedor comercial, su propio servidor de teselas o teselas sin conexión.

- **TsgcMapTileDownloader** no tiene un URLTemplate predeterminado a propósito. Apúntelo a su propio servidor o a un proveedor cuyas condiciones permitan las descargas masivas, nunca a tile.openstreetmap.org, y mantenga DelayMS por encima de cero (250 ms de forma predeterminada). Se identifica como sgcMaps-Downloader/1.0.
- Los servicios en línea envían sgcMaps/2.0 en **UserAgent** de forma predeterminada. Establézcalo con el nombre de su aplicación: Nominatim exige un agente de usuario que identifique a la aplicación.
- El componente espacia las solicitudes a Nominatim y a OpenTopoData a una por segundo, para todo el proceso, como piden sus políticas. Por eso una llamada síncrona hecha desde el hilo principal puede esperar hasta un segundo; use los métodos Async en una interfaz de usuario.
- El servidor OSRM predeterminado de TsgcMapDirections es el servidor de demostración público: solo uso ligero, una solicitud por segundo y únicamente el perfil de coche. Use su propio servidor OSRM u otro proveedor para rutas a pie y en bicicleta, y llame a **ValidateRequest** o a **IsTravelModeSupported** para comprobar un modo de viaje con el proveedor.

Los eventos asíncronos nunca llegan

Los métodos Async de los servicios ejecutan la solicitud en un hilo y disparan el evento de resultado, u OnError, en el hilo principal con TThread.Queue. Una aplicación VCL o FireMonkey entrega esos eventos desde su bucle de mensajes. Una aplicación de consola, un servicio de Windows o un hilo no tienen ese bucle, así que deben llamar a **CheckSynchronize** hasta que llegue la respuesta, y el componente debe seguir existiendo hasta entonces. Los objetos de resultado que se pasan al evento se liberan cuando el evento termina, así que copie lo que necesite dentro del manejador.

```
program GeocodeConsole;

{$APPTYPE CONSOLE}

uses
  SysUtils, Classes,
  sgcMaps_Geocoding;

type
  TListener = class
  public
    Done: Boolean;
    procedure DoResult(Sender: TObject; aResults: TsgcMapGeocodingResults);
    procedure DoError(Sender: TObject; const aError: string);
  end;

procedure TListener.DoResult(Sender: TObject;
  aResults: TsgcMapGeocodingResults);
begin
  if aResults.Count > 0 then
    WriteLn(aResults[0].DisplayName)
  else
    WriteLn('No results');
  Done := True;
end;

procedure TListener.DoError(Sender: TObject; const aError: string);
begin
  WriteLn('Error: ', aError);
  Done := True;
end;

var
  oGeocoder: TsgcMapGeocoder;
  oListener: TListener;
begin
  oListener := TListener.Create;
  oGeocoder := TsgcMapGeocoder.Create(nil);
  try
    // identify your application, as the public servers ask
    oGeocoder.UserAgent := 'MyCompanyGeocoder/1.0';
    oGeocoder.OnGeocodeResult := oListener.DoResult;
    oGeocoder.OnError := oListener.DoError;
```

```
    oGeocoder.GeocodeAsync('Plaza Mayor, Madrid');
    // a console application has no message loop, so deliver the queued
    // events with CheckSynchronize until the answer arrives
    while not oListener.Done do
      CheckSynchronize(100);
  finally
    oGeocoder.Free;
    oListener.Free;
  end;
end.
```

El **Timeout** de los servicios y del origen de teselas se aplica con THTTPClient a partir de Delphi 10.1 Berlin. En Delphi XE8 y Delphi 10 Seattle no tiene ningún efecto.

FireDAC SQLite

- Los archivos MBTiles, los archivos MBTiles vectoriales y el índice de TsgcMapOfflineGeocoder son bases de datos SQLite que se leen con FireDAC, desde Delphi XE7. Su edición de Delphi debe incluir FireDAC con el controlador SQLite, que requiere el paquete de runtime de sgcMaps.
- Las unidades crean por sí mismas el enlace del controlador FireDAC SQLite y abren los archivos en modo de solo lectura, así que el formulario no necesita ningún componente FireDAC. Al distribuir, incluya lo que necesita el controlador FireDAC SQLite en cada plataforma de destino, como describe la documentación de FireDAC de su versión de Delphi.
- Un archivo que falta genera "MBTiles file not found" o "Geocoder index not found" con la ruta. Distribuya los archivos con la aplicación y construya las rutas en tiempo de ejecución, ya que una ruta relativa depende de la carpeta actual.
- Un archivo MBTiles con otro esquema da teselas en blanco y ningún error, así que compruebe un archivo nuevo primero en una demo.
- Cada TsgcMapOfflineGeocoder tiene una conexión y debe usarse desde un solo hilo a la vez. Dé a cada hilo su propio componente. A diferencia del geocodificador en línea, genera una excepción cuando falta el índice, en lugar de devolver False.

Carpetas de caché

- Una capa en línea guarda sus teselas en **CacheFolder**. Cuando CacheFolder está vacío usa TileSourcePath, y cuando ambos están vacíos no hay caché en disco, solo una caché de 256 teselas en memoria.
- Las carpetas se crean según hace falta. Cuando no se puede escribir en la carpeta, las teselas no se guardan en caché y no se genera ningún error: esto ocurre en Program Files, y en la carpeta de la aplicación en Android e iOS. Use una carpeta del usuario, como TPath.GetCachePath en dispositivos móviles.
- Los archivos de caché se nombran solo por zoom, x e y. Dé a cada capa, proveedor y configuración de Retina una carpeta propia, o las teselas de una capa se dibujarán para otra.
- Las teselas en caché no caducan. Elimine la carpeta para descargarlas de nuevo, por ejemplo después de un cambio de estilo en el proveedor.

Pantallas de alta densidad de píxeles

El control VCL dibuja sus teselas a **Options.TileSize** píxeles, 256 de forma predeterminada, y no se escala por sí solo con los DPI del monitor. En una pantalla de alta densidad, establezca TileSize a partir de los píxeles por pulgada (de 64 a 1024) y establezca **Retina** en las capas, para que los proveedores que lo admiten envíen teselas de 512 píxeles que se mantienen nítidas al ampliarlas. Una capa vectorial con Retina dibuja sus teselas a 512 píxeles. Los proveedores que admiten teselas retina se enumeran en Teselas, capas y proveedores.

```
procedure TForm3.SetupHighDPI;
var
  oLayer: TsgcMapLayer;
begin
  oLayer := sgcVCLMap1.Layers.Add;
  oLayer.Name := 'CARTO Light';
  oLayer.Provider := mtpCartoLight;
  // request the 512 pixel (@2x) tiles of the provider
  oLayer.Retina := True;
```

```
  // a cache folder of its own: the cache file names hold only z, x and y
  oLayer.CacheFolder := ExtractFilePath(ParamStr(0)) + 'cache\carto@2x';
  // draw the map at the scale of the screen: 256 at 100%, 512 at 200%
  sgcVCLMap1.Options.TileSize := Round(256 * Screen.PixelsPerInch / 96);
end;
```

Ubicación en dispositivos móviles

- En Android, habilite el permiso **Access fine location** en las opciones del proyecto y llame a **RequestPermission** de TsgcMapLocation (Delphi 10.3 y posteriores) antes de establecer Active en True. Establecer Active en tiempo de diseño inicia el sensor antes de que el usuario haya concedido el permiso.
- En iOS y macOS, añada NSLocationWhenInUseUsageDescription a la información de versión del proyecto. En Windows, el servicio de ubicación debe estar habilitado en la configuración de privacidad.
- Los errores son "Location sensor not available", "Location sensor access denied or in error state" y "Location permission denied". Use **SimulateLocation** o **LoadTrack** para probar sin sensor, como hace la demo de FireMonkey.

Enrutamiento sin conexión

- **Open** de TsgcMapOfflineRouter genera "Invalid routing graph file", "Unsupported routing graph version" o "Truncated routing graph file" cuando GraphFile no es un archivo .sgcgraph completo de esta versión. Créelo de nuevo con la herramienta RouteGraphBuilder.
- "Waypoint n is not near a routable road" significa que el punto de paso está fuera del área del grafo, o lejos de cualquier vía del perfil seleccionado. "No route found between waypoints" significa que el grafo no tiene conexión entre ellos, lo que ocurre en el borde de un extracto.
- El grafo completo se carga en memoria, así que elija el extracto más pequeño que cubra su área.
- El lector de OpenStreetMap admite archivos .osm.pbf sin comprimir y comprimidos con zlib, el formato de los servicios de extractos habituales. Un archivo comprimido con LZMA, LZ4, Zstandard o bzip2 genera un error.

Lazarus

El control de Lazarus muestra únicamente teselas de una carpeta, porque no puede descargar teselas, y no tiene rotación ni opacidad. GeoJSON, GPX y KML no se compilan bajo FPC. Véase Lazarus.

Introducción

sgcDashboard es un dashboard nativo para Delphi y C++Builder, para VCL y FireMonkey. Un dashboard es una cuadrícula de tiles. Cada tile es un control hijo que aloja cualquier otro control colocado en él, de modo que un gráfico, un mapa, un visor de PDF, una vista previa de informe, un memo o una lista pueden convivir en la misma superficie con tema. Un control de tarjeta KPI muestra una sola cifra con su tendencia y un sparkline.
sgcDashboard forma parte de la suite visual de eSeGeCe. Sus tiles alojan los visores de **sgcCharts** (gráficos de datos en vivo), **sgcMaps** (mapas), **sgcPDF** (visor de PDF) y **sgcReports** (vista previa de informes), y se dibuja con **sgcUICore**, el paquete de runtime compartido con el lienzo, el rasterizador, las fuentes, el escritor SVG y los temas claro y oscuro. sgcDashboard requiere sgcUICore, que se instala junto a él. Los demás productos son opcionales: un tile solo los aloja cuando están instalados.
Una sola propiedad **Theme** del dashboard cambia el dashboard, sus tiles y los controles alojados entre claro y oscuro. Una build Registered permite además al usuario arrastrar y redimensionar los tiles en tiempo de ejecución y guarda la disposición como JSON, y el puente **sgcDashboardWS** alimenta las tarjetas KPI desde una conexión WebSocket, MQTT o de eventos enviados por el servidor de sgcWebSockets.

Ediciones

sgcDashboard se distribuye como Registered, Trial y Community. La edición Community omite dos características: el arrastre y redimensionado de los tiles en tiempo de ejecución con SaveLayout y LoadLayout, y el puente de datos en vivo sgcDashboardWS. Consulte Ediciones.

Por dónde empezar

- Arquitectura: el motor de disposición, los tiles, el lienzo y el tema de sgcUICore, VCL y FMX.
- Plataformas: las versiones de Delphi, C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX e Inicio rápido C++Builder: un primer dashboard en pocos minutos.
- Componentes: los componentes dashboard, tile, tarjeta KPI y origen del puente.
- Disposición, Temas, Persistencia, Datos en vivo e Imágenes KPI sin interfaz.
- Demos: los proyectos de demostración de Delphi y C++Builder.

Arquitectura

sgcDashboard está construido en capas. Cada capa solo depende de las que tiene debajo, por lo que el motor de disposición y el renderizador de KPI se compilan sin VCL ni FireMonkey.

Capa	Unidades	Función
Motor de disposición	sgcDashboard_Layout	TsgcDashboardLayout: una cuadrícula independiente del framework que convierte la celda de cada tile (Col, Row, ColSpan, RowSpan) en un rectángulo, con Columns, RowHeight, Gap y Padding en píxeles independientes del dispositivo, colocación automática, resolución de colisiones y el cálculo de ajuste para arrastrar y redimensionar. Consulte Disposición.
Dashboard y tiles	sgcDashboard_VCL, sgcDashboard_FMX	TsgcVCLDashboard y TsgcFMXDashboard poseen un motor de disposición y colocan sus tiles sobre él. TsgcVCLDashboardTile y TsgcFMXDashboardTile son controles hijos, como un TTabSheet dentro de un TPageControl, y un control colocado en un tile queda alojado en él.
Tarjeta KPI	sgcDashboard_KPI, sgcDashboard_KPI_VCL, sgcDashboard_KPI_FMX	TsgcDashboardKPI es el modelo y el renderizador. TsgcVCLKPICard y TsgcFMXKPICard son controles ligeros que poseen un modelo y lo dibujan.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme, sgcUICore_JSON	El runtime compartido: el lienzo vectorial y el rasterizador sobre los que se dibuja cada tile y cada tarjeta, la base de controles VCL y FMX con control de cadencia de fotogramas (MaxFPS) y peticiones de repintado seguras entre hilos, los tokens de los temas claro y oscuro con el aplicador de temas, y el lector y escritor de JSON que usa la persistencia de la disposición.
Puente de datos en vivo	sgcDashboardWS_Source, _WebSocket, _MQTT, _SSE	TsgcDashboardSource y sus descendientes leen un valor de cada mensaje JSON de un cliente de sgcWebSockets y lo envían a una tarjeta KPI. El puente es un paquete independiente, sgcDashboardWS, que requiere sgcDashboard y sgcWebSockets.

Los tiles son controles hijos

Un tile es un control real dentro del dashboard. En tiempo de diseño, el verbo "Add Tile" del editor de componentes del dashboard crea un tile, y cualquier control se suelta en un tile desde la paleta. Al mover o redimensionar un tile en el diseñador, este se ajusta a la cuadrícula. En tiempo de ejecución un tile se crea por código y su **Parent** se establece en el dashboard. El tile dibuja una superficie de tarjeta con borde, esquinas redondeadas y una cabecera opcional con su **Title**, y deja su área cliente a los controles que aloja.

Dibujo y temas

Los tiles y las tarjetas KPI se dibujan a través del lienzo de sgcUICore, el mismo que usa sgcCharts, de modo que un tema claro u oscuro se ve igual en ambos frameworks y en la salida PNG y SVG sin interfaz. Un tile no dibuja los controles que aloja. Les aplica el tema mediante sus propiedades publicadas: consulte Temas.

VCL y FMX

Las clases VCL y FMX tienen las mismas propiedades, métodos y eventos. El dashboard VCL desciende de TScrollingWinControl y el dashboard FMX de TVertScrollBox, por lo que una cuadrícula más alta que la vista se des-

plaza verticalmente. Todos los tamaños de la disposición están en píxeles independientes del dispositivo y se escalan según el PPI del monitor en VCL y según la escala del formulario en FMX.

Véase también

- Plataformas
- Disposición
- Temas

Ediciones

sgcDashboard se construye a partir del mismo código fuente para todas las ediciones. La edición solo cambia las características que compila una build.

Edición	Incluye
Registered	Todo: los dashboards y tiles VCL y FMX, las tarjetas KPI y las imágenes KPI sin interfaz, los temas, el arrastre y redimensionado de los tiles en tiempo de ejecución con SaveLayout y LoadLayout, y el puente de datos en vivo sgcDashboardWS.
Trial	Las mismas características que Registered. Una build compilada con SGC_TRIAL muestra un aviso de que es una versión de prueba una vez superada la ventana de versiones indicada en la build. Nada más cambia.
Community	Los dashboards y tiles, la disposición en cuadrícula con colocación automática, las tarjetas KPI con sparklines y salida PNG y SVG sin interfaz, y los temas claro y oscuro. No incluye el arrastre y redimensionado en tiempo de ejecución con SaveLayout y LoadLayout, ni el puente sgcDashboardWS.

Qué no incluye Community

La edición Community se compila con SGC_DASHBOARD_PERSIST sin definir. Estos miembros no existen en una build Community:

- **AllowDrag**, **AllowResize** y **OnLayoutChanged** en el dashboard.
- **SaveLayout**, **LoadLayout**, **SaveLayoutToFile**, **LoadLayoutFromFile**, **SaveLayoutToStream** y **LoadLayoutFromStream** en el dashboard.
- **Locked** en el tile VCL. En FMX, Locked es la propiedad estándar de FMX y sigue disponible, simplemente no hay arrastre que bloquear.

La edición Community también se distribuye sin las unidades y paquetes de sgcDashboardWS, por lo que TsgcDashboardSource_WebSocket, TsgcDashboardSource_MQTT y TsgcDashboardSource_SSE no están disponibles. Una tarjeta KPI se sigue alimentando por código con **Value** y **AddValue**.
La edición es una decisión en tiempo de compilación en sgcVer.inc (SGC_COMMUNITY, que también desactiva SGC_TRIAL), no un interruptor en tiempo de ejecución. El código que usa alguno de los miembros anteriores no compila contra una build Community, consulte Solución de problemas.

Véase también

- Persistencia
- Datos en vivo
- Plataformas

Instalación

sgcDashboard forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcDashboard** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcDashboard

- **sgcUICore**, la base compartida de los productos de disposición y de sgcCharts. No tiene casilla propia. El instalador la instala automáticamente siempre que se selecciona sgcDashboard.
- **sgcDashboard para sgcWebSockets**, que aparece en **sgcWebSockets bridges**. Instala los paquetes sgcDashboardWS con los orígenes WebSocket, MQTT y SSE (consulte Datos en vivo). El instalador solo lo ofrece cuando sgcWebSockets ya está instalado en el equipo y su licencia incluye sgcDashboard. De lo contrario aparece atenuado.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Dashboard`	Las demos de sgcDashboard. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluido sgcDashboard.
- **Community**: el instalador gratuito Community de sgcUI incluye sgcDashboard, junto con sgcCharts, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard. Los puentes de sgcWebSockets no forman parte del instalador Community, y sgcPDF y sgcMaps no forman parte de Community. Consulte Ediciones para saber qué deja fuera la compilación Community.

Instalaciones independientes anteriores

Si el instalador encuentra una instalación independiente anterior de sgcUICore, como la que escribió el instalador independiente de sgcCharts, muestra las rutas de bibliotecas y los paquetes de diseño de esa instalación en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcDashboardD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcDashboard<V>` / `dclsgcDashboard<V>`	Los componentes de sgcDashboard (tiempo de ejecución / tiempo de diseño).
`sgcDashboardWS<V>` / `dclsgcDashboardWS<V>`	El puente de sgcWebSockets (tiempo de ejecución / tiempo de diseño). Requiere sgcWebSockets.

Compílelos en orden de dependencia: `sgcUICore`, después `sgcDashboard`, después `dclsgcDashboard` y, por último, los paquetes del puente. Después instale los paquetes de diseño `dcl*` y añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` correspondiente a su versión, por ejemplo `sgcDashboardC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

.NET

No cambia nada para .NET. sgcDashboard se sigue distribuyendo como sus propios paquetes NuGet (`esegece.sgcDashboard` y sus paquetes complementarios). Consulte la instalación de .NET.

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13. Los paquetes generan las cabeceras .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	TsgcVCLDashboard, TsgcVCLDashboardTile y TsgcVCLKPICard en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	TsgcFMXDashboard, TsgcFMXDashboardTile y TsgcFMXKPICard desde Delphi XE6 (SGC_DASHBOARD_FMX), en Win32, Win64, macOS, iOS y Android.
Linux64	Las unidades independientes del framework, el motor de disposición (sgcDashboard_Layout) y el modelo KPI con su salida PNG y SVG sin interfaz (sgcDashboard_KPI), desde Delphi 10.2 (SGC_DASHBOARD_LINUX). Los controles FMX en Linux necesitan FMX para Linux (FMXLinux).
Puente de datos en vivo	Los paquetes sgcDashboardWS, donde estén instalados los paquetes de sgcWebSockets para el mismo IDE.
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, la salida sin interfaz en PNG y SVG y el puente de sgcWebSockets, en .NET Framework 4.7.2 y .NET 8 y 9, véase Visión general de .NET.

Paquetes

sgcDashboard incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcDashboardD13.dpk para Delphi 13 y sgcDashboardC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcDashboard	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, sgcUICore
sgcDashboardVCL	Runtime, VCL	sgcDashboard_VCL y sgcDashboard_KPI_VCL (TsgcVCLDashboard, TsgcVCLDashboardTile, TsgcVCLKPICard)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcDashboard
sgcDashboardFMX	Runtime, FireMonkey, desde XE6	sgcDashboard_FMX y sgcDashboard_KPI_FMX (TsgcFMXDashboard, TsgcFMXDashboardTile, TsgcFMXKPICard)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcDashboard
dclsgcDashboard	Tiempo de diseño, VCL	Registra los componentes VCL en la página de paleta **SGC Dashboard** y añade el verbo "Add Tile" al editor de componentes del dashboard.	designide, rtl, vcl, sgcDashboard, sgcDashboardVCL
dclsgcDashboardFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra los componentes FireMonkey en la página de paleta **SGC Dashboard**, con el mismo verbo "Add Tile".	designide, rtl, fmx, sgcDashboard, sgcDashboardFMX, dclsgcDashboard

sgcDash-boardWS, dclsgcDash-boardWS	Runtime y tiempo de di-seño, puente de sgcWeb-Sockets	El puente de datos en vivo, independiente del framework.	rtl, sgcDashboard, sgcWebSockets

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcDashboardFMX requiere dclsgcDashboard, así que instale primero dclsgcDashboard.

Versiones del IDE y plataformas de destino

Delphi	sgcDashboard	sgcDash-boardVCL	sgcDashboardFMX	dclsgcDashboard, dclsgcDash-boardFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Li-nux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Li-nux64, OSX64, An-droid, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, An-droid64, iOSDevice64	Win32
11	Win32, Win64, Li-nux64, OSX64, OS-XARM64, Android, Android64, iOSDevi-ce64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSi-mARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcDashboard, sgcDash-boardVCL	sgcDash-boardFMX	dclsgcDashboard, dclsgcDash-boardFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcDashboard)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcDashboard)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcDashboardC13.cbproj, sgcDashboardVCLC13.cbproj, sgcDashboardFMXC13.cbproj, dclsgcDashboardC13.cbproj y dclsgcDashboardFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64,

nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXDashboard, TsgcFMXDashboardTile y TsgcFMXKPICard) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcDashboardVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcDashboard_FMX y sgcDashboard_KPI_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcDashboard se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa el motor de disposición y el modelo KPI con su salida PNG y SVG sin interfaz. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Capturar un formulario VCL con PaintTo

Para capturar un formulario VCL, o un dashboard en él, en un mapa de bits con PaintTo, bloquee el canvas del mapa de bits alrededor de la llamada. PaintTo sobre un canvas sin bloquear puede perder su contexto de dispositivo cuando un control nativo del formulario procesa un mensaje durante la captura, y el mapa de bits queda en blanco o incompleto. Un formulario FireMonkey usa MakeScreenshot en su lugar.

```
procedure CaptureForm(aForm: TForm; const aFileName: string);
var
  vBitmap: TBitmap;
begin
  vBitmap := TBitmap.Create;
  try
    vBitmap.SetSize(aForm.ClientWidth, aForm.ClientHeight);
    vBitmap.Canvas.Lock;
    try
      aForm.PaintTo(vBitmap.Canvas, 0, 0);
    finally
      vBitmap.Canvas.Unlock;
    end;
    vBitmap.SaveToFile(aFileName);
  finally
    vBitmap.Free;
  end;
end;
```

Véase también

- Ediciones
- Arquitectura
- Solución de problemas

Inicio rápido VCL

En tiempo de diseño

- Suelte un **TsgcVCLDashboard** desde la página de paleta **SGC Dashboard** en un formulario y establezca **Align** en alClient.
- Haga clic con el botón derecho en el dashboard y elija **Add Tile**. Repítalo para cada tile. Un tile nuevo tiene Col y Row en -1, así que la colocación automática lo sitúa en el primer hueco libre de la cuadrícula. Establezca Col, Row, ColSpan y RowSpan en el Inspector de Objetos, o mueva y redimensione el tile en el diseñador, para darle una celda fija.
- Seleccione un tile y suelte un control en él, por ejemplo un TsgcVCLChart de sgcCharts o un TMemo, y establezca su **Align** en alClient.
- Suelte un **TsgcVCLKPICard** en otro tile, establezca su Align en alClient y rellene **Title**, **Value**, **Trend** y **TrendValue**. Establezca **ShowHeader** de ese tile en False, ya que la tarjeta muestra su propio título.
- Establezca **Theme** del dashboard en utDark. El dashboard, los tiles, la tarjeta KPI y los controles alojados pasan a la paleta oscura.

En código

El mismo dashboard creado en tiempo de ejecución. El programa compila tal cual con `sgcUI\Source` en la ruta de búsqueda.

```
uses
  Forms, Classes, Controls, StdCtrls,
  sgcUICore_Theme, sgcDashboard_KPI, sgcDashboard_VCL, sgcDashboard_KPI_VCL;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oTile: TsgcVCLDashboardTile;
  oCard: TsgcVCLKPICard;
  oMemo: TMemo;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcDashboard VCL Quick Start';
  Width := 900;
  Height := 560;

  // the dashboard: a 12 column grid that fills the form
  FDashboard := TsgcVCLDashboard.Create(Self);
  FDashboard.Parent := Self;
  FDashboard.Align := alClient;
  FDashboard.Columns := 12;
  FDashboard.RowHeight := 110;

  // a KPI tile at column 0, row 0, 3 columns wide and 2 rows high
  oTile := TsgcVCLDashboardTile.Create(Self);
  oTile.Name := 'tileRevenue';
  oTile.Parent := FDashboard;
  oTile.Col := 0;
  oTile.Row := 0;
  oTile.ColSpan := 3;
  oTile.RowSpan := 2;
  oTile.ShowHeader := False;

  oCard := TsgcVCLKPICard.Create(Self);
  oCard.Parent := oTile;
  oCard.Align := alClient;
  oCard.Title := 'Revenue';
  oCard.Value := '12,480 USD';
  oCard.Icon := '$';
  oCard.Trend := ktUp;
```

```
  oCard.TrendValue := '+4.2 %';
  oCard.FooterText := 'vs last week';
  oCard.AddValue(11200);
  oCard.AddValue(11900);
  oCard.AddValue(12480);

  // a log tile, Col and Row stay -1: auto-flow puts it in the first free slot
  oTile := TsgcVCLDashboardTile.Create(Self);
  oTile.Name := 'tileEvents';
  oTile.Parent := FDashboard;
  oTile.Title := 'Events';
  oTile.ColSpan := 6;
  oTile.RowSpan := 3;

  oMemo := TMemo.Create(Self);
  oMemo.Parent := oTile;
  oMemo.Align := alClient;
  oMemo.Lines.Add('Service started');

  // one property themes the dashboard, the tiles and the hosted controls
  FDashboard.Theme := utDark;
end;
```

Véase también

- TsgcVCLDashboard
- TsgcVCLDashboardTile
- TsgcVCLKPICard
- Disposición
- Temas

Inicio rápido FMX

Los controles de FireMonkey necesitan Delphi XE6 o posterior. Se usan exactamente igual que los de VCL: suelte un **TsgcFMXDashboard**, establezca **Align** en Client, use el verbo **Add Tile** y suelte controles o un **TsgcFMXKPICard** en los tiles.

En código

```
uses
  System.Classes, FMX.Forms, FMX.Types,
  sgcUICore_Theme, sgcDashboard_KPI, sgcDashboard_FMX, sgcDashboard_KPI_FMX;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oTile: TsgcFMXDashboardTile;
  oCard: TsgcFMXKPICard;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcDashboard FMX Quick Start';
  Width := 900;
  Height := 560;

  FDashboard := TsgcFMXDashboard.Create(Self);
  FDashboard.Parent := Self;
  FDashboard.Align := TAlignLayout.Client;
  FDashboard.Columns := 12;
  FDashboard.RowHeight := 110;

  oTile := TsgcFMXDashboardTile.Create(Self);
  oTile.Parent := FDashboard;
  oTile.ColSpan := 4;
  oTile.RowSpan := 2;
  oTile.ShowHeader := False;

  oCard := TsgcFMXKPICard.Create(Self);
  oCard.Parent := oTile;
  oCard.Align := TAlignLayout.Client;
  oCard.Title := 'Active users';
  oCard.Value := '1,284';
  oCard.Trend := ktDown;
  oCard.TrendValue := '-2.1 %';
  oCard.AddValue(1310);
  oCard.AddValue(1284);

  FDashboard.Theme := utDark;
end;
```

Véase también

- TsgcFMXDashboard
- TsgcFMXDashboardTile
- TsgcFMXKPICard
- Plataformas

Inicio rápido C++Builder

Instale los paquetes de sgcDashboard para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de inclusión y de bibliotecas del proyecto, y use los componentes de la página de paleta **SGC Dashboard** exactamente igual que en Delphi. El verbo **Add Tile** funciona de la misma manera. El instalador añade ambas rutas al IDE por usted. Los cabeceras .hpp y los archivos .lib generados de sgcDashboard y sgcUICore están en esa única carpeta, por lo que sgcUICore no necesita una ruta propia.

En código

Véase también

- Inicio rápido VCL
- Demos
- Plataformas

Demos

Las demos están en la carpeta **Demos** de la instalación, con un proyecto Delphi en Demos\Delphi y la misma demo para C++Builder en Demos\CBuilder. Cada carpeta de demo tiene un readme.html que explica qué hace la demo y cómo funciona. Las demos son aplicaciones VCL para Win32 y Win64.

Demo	Componentes	Qué muestra
Basic	TsgcVCLDashboard, TsgcVCLDashboardTile, TsgcVCLKPICard	Un dashboard con seis tiles en una cuadrícula de 12 columnas: cuatro tarjetas KPI en vivo movidas por un temporizador, un registro de eventos y una lista de servidores, con un interruptor de tema claro y oscuro que también aplica el tema al TMemo y al TListBox.
Suite	TsgcVCLDashboard, TsgcVCLKPICard, TsgcVCLChart, TsgcVCLMap, TsgcPDFViewerControl, TsgcReportPreviewVCL	Un dashboard cuyos tiles alojan los demás productos VCL de eSeGeCe: un gráfico de líneas en vivo de sgcCharts, un mapa de sgcMaps, un visor de sgcPDF y una vista previa de sgcReports, junto a dos tarjetas KPI, todo con el mismo tema. Necesita los fuentes de sgcCharts, sgcMaps, sgcPDF y sgcReports en la ruta de búsqueda.
LiveKPI	TsgcVCLKPICard, TsgcDashboardSource_WebSocket, TsgcWebSocketClient, TsgcWebSocketServer	Tres tarjetas KPI enlazadas a un flujo WebSocket mediante TsgcDashboardSource_WebSocket. Un TsgcWebSocketServer dentro de la demo emite un mensaje JSON cada 500 ms, y cada origen lee su propio ValuePath de él. Necesita los fuentes de sgcWebSockets y no forma parte de la edición Community.
SaveLayout	TsgcVCLDashboard, TsgcVCLDashboardTile, TsgcVCLKPICard	Un dashboard que el usuario reorganiza en tiempo de ejecución con AllowDrag y AllowResize, una disposición guardada y cargada desde layout.json con SaveLayoutToFile y LoadLayoutFromFile, y OnLayoutChanged. No forma parte de la edición Community.

Véase también

- Inicio rápido VCL
- Persistencia
- Datos en vivo

TsgcVCLDashboard

Unidad: sgcDashboard_VCL
TsgcVCLDashboard es el dashboard VCL, solo para Windows. Organiza sus hijos TsgcVCLDashboardTile en una cuadrícula de **Columns** columnas y filas de **RowHeight** píxeles, con **Gap** entre tiles y **Padding** alrededor, y se desplaza verticalmente cuando los tiles son más altos que el control. Los tiles se añaden en tiempo de diseño con el verbo **Add Tile** del editor de componentes. Consulte Disposición para saber cómo se colocan las celdas.

Propiedades

Propiedad	Tipo	Descripción
Theme	TsgcUITheme	utLight (predeterminado), utDark o utCustom. Dibuja el fondo del dashboard y lo siguen todos los tiles cuyo ParentTheme es True, que lo aplican a los controles que alojan. Consulte Temas.
Columns	Integer	Número de columnas de la cuadrícula. Predeterminado 12.
RowHeight	Integer	Altura de una fila de la cuadrícula, en píxeles independientes del dispositivo. Predeterminado 120.
Gap	Integer	Espacio entre dos tiles, en píxeles independientes del dispositivo. Predeterminado 12.
Padding	Integer	Espacio entre el borde del dashboard y los tiles, en píxeles independientes del dispositivo. Predeterminado 12.
AutoFlow	Boolean	Cuando es True (predeterminado), un tile cuyo Col o Row es -1 ocupa el primer hueco libre de la cuadrícula. Cuando es False, -1 se interpreta como 0.
AllowDrag	Boolean	Movimiento de un tile en tiempo de ejecución arrastrando su cabecera. Predeterminado False. No está en la edición Community.
AllowResize	Boolean	Redimensionado de un tile en tiempo de ejecución con el asa de su esquina inferior derecha. Predeterminado False. No está en la edición Community.
Layout	TsgcDashboardLayout (public, solo lectura)	El motor de disposición de la cuadrícula del dashboard. Consulte Disposición.

También se publican los miembros habituales de TScrollingWinControl: **Align**, **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop**, **Visible**, **OnClick**, **OnDblClick**, **OnMouseDown**, **OnMouseMove**, **OnMouseUp** y **OnResize**.

Métodos

Método	Devuelve	Descripción

Arrange		Coloca los tiles en la cuadrícula ahora. El dashboard se reorganiza solo cuando cambia un tile o una propiedad de la cuadrícula, por lo que rara vez hace falta llamarlo.
ApplyTheme		Vuelve a aplicar el tema de cada tile a los controles que aloja, por ejemplo después de añadir o cambiar controles por código.
SaveLayout	string	Las métricas de la cuadrícula y la celda y la visibilidad de cada tile con nombre, como JSON. No está en la edición Community.
LoadLayout(const aJSON: string)		Aplica un texto de SaveLayout. Los tiles se emparejan por Name, los nombres desconocidos se ignoran y los tiles que faltan en el texto conservan sus celdas. Una sola reorganización y después OnLayoutChanged. No está en la edición Community.
SaveLayoutToFile(const aFileName: string)		Escribe SaveLayout en un archivo, UTF-8 sin BOM. No está en la edición Community.
LoadLayoutFromFile(const aFileName: string)		Lee un archivo de disposición escrito por SaveLayoutToFile. Se omite un BOM. No está en la edición Community.
SaveLayoutToStream(aStream: TStream)		Escribe SaveLayout en un flujo, UTF-8 sin BOM. No está en la edición Community.
LoadLayoutFromStream(aStream: TStream)		Lee una disposición desde un flujo. Se omite un BOM. No está en la edición Community.

Eventos

Evento	Tipo	Descripción
OnLayoutChanged	TNotifyEvent	Un arrastre o redimensionado en tiempo de ejecución cambió la celda de un tile, o se aplicó LoadLayout. No está en la edición Community.

Ejemplo

```
constructor TfrmMain.Create(aOwner: TComponent);
var
  oTile: TsgcVCLDashboardTile;
  oCard: TsgcVCLKPICard;
  oMemo: TMemo;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcDashboard VCL Quick Start';
  Width := 900;
  Height := 560;

  // the dashboard: a 12 column grid that fills the form
  FDashboard := TsgcVCLDashboard.Create(Self);
  FDashboard.Parent := Self;
  FDashboard.Align := alClient;
  FDashboard.Columns := 12;
  FDashboard.RowHeight := 110;

  // a KPI tile at column 0, row 0, 3 columns wide and 2 rows high
  oTile := TsgcVCLDashboardTile.Create(Self);
  oTile.Name := 'tileRevenue';
```

```
  oTile.Parent := FDashboard;
  oTile.Col := 0;
  oTile.Row := 0;
  oTile.ColSpan := 3;
  oTile.RowSpan := 2;
  oTile.ShowHeader := False;

  oCard := TsgcVCLKPICard.Create(Self);
  oCard.Parent := oTile;
  oCard.Align := alClient;
  oCard.Title := 'Revenue';
  oCard.Value := '12,480 USD';
  oCard.Icon := '$';
  oCard.Trend := ktUp;
  oCard.TrendValue := '+4.2 %';
  oCard.FooterText := 'vs last week';
  oCard.AddValue(11200);
  oCard.AddValue(11900);
  oCard.AddValue(12480);

  // a log tile, Col and Row stay -1: auto-flow puts it in the first free slot
  oTile := TsgcVCLDashboardTile.Create(Self);
  oTile.Name := 'tileEvents';
  oTile.Parent := FDashboard;
  oTile.Title := 'Events';
  oTile.ColSpan := 6;
  oTile.RowSpan := 3;

  oMemo := TMemo.Create(Self);
  oMemo.Parent := oTile;
  oMemo.Align := alClient;
  oMemo.Lines.Add('Service started');

  // one property themes the dashboard, the tiles and the hosted controls
  FDashboard.Theme := utDark;
end;
```

Véase también

- TsgcVCLDashboardTile
- TsgcFMXDashboard
- Disposición
- Persistencia
- Temas

TsgcVCLDashboardTile

Unidad: sgcDashboard_VCL
TsgcVCLDashboardTile es un tile de un TsgcVCLDashboard. Dibuja una tarjeta con tema, con borde, esquinas redondeadas y una cabecera opcional, y aloja cualquier control VCL colocado en ella: un gráfico, un mapa, un visor de PDF, una vista previa de informe, una tarjeta KPI o un control estándar. El tile aplica su tema efectivo a los controles que aloja. El tile nunca dibuja por sí mismo los controles alojados.

Propiedades

Propiedad	Tipo	Descripción
Title	string	Texto de la cabecera del tile.
ShowHeader	Boolean	Dibuja la banda de cabecera con el Title. Predeterminado True. El área cliente del tile empieza debajo de la cabecera.
Col	Integer	Columna de la cuadrícula del tile, basada en cero. Predeterminado -1, colocado por la colocación automática.
Row	Integer	Fila de la cuadrícula del tile, basada en cero. Predeterminado -1, colocado por la colocación automática.
ColSpan	Integer	Número de columnas que cubre el tile. Predeterminado 3, limitado de 1 a Columns.
RowSpan	Integer	Número de filas que cubre el tile. Predeterminado 2, como mínimo 1.
ParentTheme	Boolean	El tile y sus controles alojados siguen el Theme del dashboard. Predeterminado True.
Theme	TsgcUITheme	Tema del tile. Al establecerlo se desactiva ParentTheme. Fuera de un dashboard, un tile con ParentTheme usa utLight.
Locked	Boolean	El tile nunca se mueve ni se redimensiona en tiempo de ejecución, aunque el dashboard lo permita. Predeterminado False. No está en la edición Community.
Dashboard	TsgcVCLDashboard (public, solo lectura)	El dashboard al que pertenece el tile, nil cuando su padre no es un dashboard.

También se publican los miembros habituales de un control: **Constraints**, **Enabled**, **Hint**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **Visible**, **OnClick**, **OnContextPopup**, **OnDblClick**, **OnMouseDown**, **OnMouseMove**, **OnMouseUp** y **OnResize**. Un tile oculto (Visible False) deja libre su celda.

Métodos

Método	Devuelve	Descripción
GetCell	TsgcDashboardCell	La celda solicitada del tile, un registro con Col, Row, ColSpan y RowSpan.

SetCell(const aCell: TsgcDash-boardCell)		Establece Col, Row, ColSpan y RowSpan a la vez, con una única reorganización del dashboard.

Eventos

Evento	Tipo	Descripción
OnResi-ze	TNotifyE-vent	El tile se redimensionó, por una reorganización del dashboard o por un redimen-sionado en tiempo de ejecución.

Ejemplo

Los tiles siguen el tema del dashboard salvo que establezcan el suyo:

```
procedure ApplyThemes(aDashboard: TsgcVCLDashboard;
  aTileLight: TsgcVCLDashboardTile);
begin
  // every tile with ParentTheme = True follows the dashboard
  aDashboard.Theme := utDark;
  // this tile keeps its own theme: setting Theme clears ParentTheme
  aTileLight.Theme := utLight;
  // back to the dashboard theme
  aTileLight.ParentTheme := True;
  // pushes the theme again, after controls were changed in code
  aDashboard.ApplyTheme;
end;
```

Véase también

- TsgcVCLDashboard
- TsgcFMXDashboardTile
- Disposición
- Temas

TsgcFMXDashboard

Unidad: sgcDashboard_FMX
TsgcFMXDashboard es el dashboard de FireMonkey, disponible desde Delphi XE6. Tiene los mismos miembros de cuadrícula, tema y persistencia que TsgcVCLDashboard. Desciende de TVertScrollBox, por lo que los tiles se desplazan verticalmente, y su **Padding** es un Integer en píxeles independientes del dispositivo que sustituye al Padding de tipo TBounds de los controles FMX. El verbo **Add Tile** del editor de componentes está disponible en el IDE de Windows.

Propiedades

Propiedad	Tipo	Descripción
Theme	TsgcUITheme	utLight (predeterminado), utDark o utCustom. Dibuja el fondo del dashboard y lo siguen todos los tiles cuyo ParentTheme es True, que lo aplican a los controles que alojan. Consulte Temas.
Columns	Integer	Número de columnas de la cuadrícula. Predeterminado 12.
RowHeight	Integer	Altura de una fila de la cuadrícula, en píxeles independientes del dispositivo. Predeterminado 120.
Gap	Integer	Espacio entre dos tiles, en píxeles independientes del dispositivo. Predeterminado 12.
Padding	Integer	Espacio entre el borde del dashboard y los tiles, en píxeles independientes del dispositivo. Predeterminado 12.
AutoFlow	Boolean	Cuando es True (predeterminado), un tile cuyo Col o Row es -1 ocupa el primer hueco libre de la cuadrícula. Cuando es False, -1 se interpreta como 0.
AllowDrag	Boolean	Movimiento de un tile en tiempo de ejecución arrastrando su cabecera. Predeterminado False. No está en la edición Community.
AllowResize	Boolean	Redimensionado de un tile en tiempo de ejecución con el asa de su esquina inferior derecha. Predeterminado False. No está en la edición Community.
Layout	TsgcDashboardLayout (public, solo lectura)	El motor de disposición de la cuadrícula del dashboard. Consulte Disposición.

Métodos

Método	Devuelve	Descripción
Arrange		Coloca los tiles en la cuadrícula ahora. El dashboard se reorganiza solo cuando cambia un tile o una propiedad de la cuadrícula, por lo que rara vez hace falta llamarlo.

ApplyTheme		Vuelve a aplicar el tema de cada tile a los controles que aloja, por ejemplo después de añadir o cambiar controles por código.
SaveLayout	string	Las métricas de la cuadrícula y la celda y la visibilidad de cada tile con nombre, como JSON. No está en la edición Community.
LoadLayout(const aJSON: string)		Aplica un texto de SaveLayout. Los tiles se emparejan por Name, los nombres desconocidos se ignoran y los tiles que faltan en el texto conservan sus celdas. Una sola reorganización y después OnLayoutChanged. No está en la edición Community.
SaveLayoutToFile(const aFileName: string)		Escribe SaveLayout en un archivo, UTF-8 sin BOM. No está en la edición Community.
LoadLayoutFromFile(const aFileName: string)		Lee un archivo de disposición escrito por SaveLayoutToFile. Se omite un BOM. No está en la edición Community.
SaveLayoutToStream(aStream: TStream)		Escribe SaveLayout en un flujo, UTF-8 sin BOM. No está en la edición Community.
LoadLayoutFromStream(aStream: TStream)		Lee una disposición desde un flujo. Se omite un BOM. No está en la edición Community.

Eventos

Evento	Tipo	Descripción
OnLayoutChanged	TNotifyEvent	Un arrastre o redimensionado en tiempo de ejecución cambió la celda de un tile, o se aplicó LoadLayout. No está en la edición Community.

Ejemplo

```
constructor TfrmMain.Create(aOwner: TComponent);
var
  oTile: TsgcFMXDashboardTile;
  oCard: TsgcFMXKPICard;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcDashboard FMX Quick Start';
  Width := 900;
  Height := 560;

  FDashboard := TsgcFMXDashboard.Create(Self);
  FDashboard.Parent := Self;
  FDashboard.Align := TAlignLayout.Client;
  FDashboard.Columns := 12;
  FDashboard.RowHeight := 110;

  oTile := TsgcFMXDashboardTile.Create(Self);
  oTile.Parent := FDashboard;
  oTile.ColSpan := 4;
  oTile.RowSpan := 2;
  oTile.ShowHeader := False;

  oCard := TsgcFMXKPICard.Create(Self);
  oCard.Parent := oTile;
  oCard.Align := TAlignLayout.Client;
```

```
  oCard.Title := 'Active users';
  oCard.Value := '1,284';
  oCard.Trend := ktDown;
  oCard.TrendValue := '-2.1 %';
  oCard.AddValue(1310);
  oCard.AddValue(1284);

  FDashboard.Theme := utDark;
end;
```

Véase también

- TsgcFMXDashboardTile
- TsgcVCLDashboard
- Inicio rápido FMX

TsgcFMXDashboardTile

Unidad: sgcDashboard_FMX
TsgcFMXDashboardTile es un tile de un TsgcFMXDashboard, con los mismos miembros que TsgcVCLDashboardTile. Un tile cuya propiedad estándar de FMX **Locked** es True nunca inicia un arrastre ni un redimensionado en tiempo de ejecución.

Propiedades

Propiedad	Tipo	Descripción
Title	string	Texto de la cabecera del tile.
ShowHeader	Boolean	Dibuja la banda de cabecera con el Title. Predeterminado True. El área cliente del tile empieza debajo de la cabecera.
Col	Integer	Columna de la cuadrícula del tile, basada en cero. Predeterminado -1, colocado por la colocación automática.
Row	Integer	Fila de la cuadrícula del tile, basada en cero. Predeterminado -1, colocado por la colocación automática.
ColSpan	Integer	Número de columnas que cubre el tile. Predeterminado 3, limitado de 1 a Columns.
RowSpan	Integer	Número de filas que cubre el tile. Predeterminado 2, como mínimo 1.
ParentTheme	Boolean	El tile y sus controles alojados siguen el Theme del dashboard. Predeterminado True.
Theme	TsgcUITheme	Tema del tile. Al establecerlo se desactiva ParentTheme. Fuera de un dashboard, un tile con ParentTheme usa utLight.
Locked	Boolean	La propiedad estándar de FMX. Un tile bloqueado nunca se mueve ni se redimensiona en tiempo de ejecución. Predeterminado False.
Dashboard	TsgcFMXDashboard (public, solo lectura)	El dashboard al que pertenece el tile, nil cuando su padre no es un dashboard.

También se publican los miembros habituales de un control FMX, entre ellos **Anchors**, **Enabled**, **Hint**, **Margins**, **Opacity**, **PopupMenu**, **Visible**, y los eventos de ratón, de dibujo y de redimensionado.

Métodos

Método	Devuelve	Descripción
GetCell	TsgcDashboardCell	La celda solicitada del tile, un registro con Col, Row, ColSpan y RowSpan.
SetCell(const aCell: TsgcDashboardCell)		Establece Col, Row, ColSpan y RowSpan a la vez, con una única reorganización del dashboard.

Eventos

Evento	Tipo	Descripción
OnResize	TNotifyEvent	El tile se redimensionó.
OnResized	TNotifyEvent	Terminó el redimensionado del tile.

Ejemplo

```
constructor TfrmMain.Create(aOwner: TComponent);
var
  oTile: TsgcFMXDashboardTile;
  oCard: TsgcFMXKPICard;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcDashboard FMX Quick Start';
  Width := 900;
  Height := 560;

  FDashboard := TsgcFMXDashboard.Create(Self);
  FDashboard.Parent := Self;
  FDashboard.Align := TAlignLayout.Client;
  FDashboard.Columns := 12;
  FDashboard.RowHeight := 110;

  oTile := TsgcFMXDashboardTile.Create(Self);
  oTile.Parent := FDashboard;
  oTile.ColSpan := 4;
  oTile.RowSpan := 2;
  oTile.ShowHeader := False;

  oCard := TsgcFMXKPICard.Create(Self);
  oCard.Parent := oTile;
  oCard.Align := TAlignLayout.Client;
  oCard.Title := 'Active users';
  oCard.Value := '1,284';
  oCard.Trend := ktDown;
  oCard.TrendValue := '-2.1 %';
  oCard.AddValue(1310);
  oCard.AddValue(1284);

  FDashboard.Theme := utDark;
end;
```

Véase también

- TsgcFMXDashboard
- TsgcVCLDashboardTile
- Temas

TsgcDashboardKPI

Unidad: sgcDashboard_KPI
TsgcDashboardKPI es el modelo y renderizador independiente del framework de una tarjeta KPI: un título, un valor grande, un icono opcional, una flecha de tendencia con su texto, un texto de pie y un sparkline con los últimos MaxPoints valores. **Render** dibuja la tarjeta en cualquier lienzo de sgcUICore, de modo que el mismo código dibuja TsgcVCLKPICard, TsgcFMXKPICard y las imágenes PNG y SVG sin interfaz. Se compila en todas las plataformas, incluida Linux64.
AddValue y ClearValues pueden llamarse desde cualquier hilo, los valores están protegidos por un bloqueo y OnChange se invoca entonces desde ese hilo. Las demás propiedades están pensadas para establecerse desde el hilo principal.

Propiedades

Propiedad	Tipo	Descripción
Title	string	Título en la parte superior de la tarjeta.
Value	string	La cifra grande de la tarjeta, ya formateada.
Icon	string	Un glifo de texto corto dibujado en un círculo de acento, por ejemplo '$' o '%'. Si está vacío no se dibuja icono.
Trend	TsgcDashboardKPITrend	ktNone (predeterminado), ktUp, ktDown o ktNeutral. Dibuja la flecha de tendencia, verde para subida y roja para bajada.
TrendValue	string	Texto junto a la flecha de tendencia, por ejemplo '+4.2 %'.
FooterText	string	Texto pequeño en la parte inferior de la tarjeta.
AccentColor	TsgcUIColor	Color $AARRGGBB del círculo del icono y del sparkline. 0 (predeterminado) usa el color de acento del tema.
Theme	TsgcUITheme	utLight (predeterminado) o utDark. Dentro de un tile de dashboard lo establece el tile.
MaxPoints	Integer	Número de valores que conserva el sparkline. Predeterminado 50. El valor más antiguo se descarta cuando llega uno nuevo.
ValueCount	Integer (public, solo lectura)	Número de valores del sparkline.
Values[Index: Integer]	Double (public, solo lectura)	Un valor del sparkline, del más antiguo al más reciente.

Métodos

Método	Devuelve	Descripción
AddValue(aValue: Double)		Añade un valor al sparkline. Seguro entre hilos. Los valores NaN e infinitos se ignoran.
ClearValues		Elimina todos los valores del sparkline.

GetValueArray	TsgcDashboardKPIValues	Una copia de los valores del sparkline, del más antiguo al más reciente.
Render(aCanvas: TsgcUICanvas; aWidth, aHeight: Integer; aScale: Double = 1)		Dibuja la tarjeta en aCanvas, con aWidth por aHeight píxeles de dispositivo. aScale es el factor HiDPI aplicado a todos los tamaños.
SaveToPNG(const aFileName: string; aWidth, aHeight: Integer; aScale: Double = 1)		Renderiza la tarjeta en un archivo PNG. aWidth y aHeight son píxeles lógicos, la imagen mide Round(aWidth * aScale) por Round(aHeight * aScale) píxeles. Una sobrecarga escribe en un TStream.
SaveToSVG(const aFileName: string; aWidth, aHeight: Integer; aScale: Double = 1)		Renderiza la tarjeta en un archivo SVG. Una sobrecarga escribe en un TStream.
Assign(Source: TPersistent)		Copia las propiedades y los valores del sparkline de otro TsgcDashboardKPI.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Cambió una propiedad o el sparkline. Se invoca en el hilo que hizo el cambio.

Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcDashboard_KPI;

procedure SaveKPIImages;
var
  oKPI: TsgcDashboardKPI;
begin
  oKPI := TsgcDashboardKPI.Create;
  try
    oKPI.Title := 'Orders';
    oKPI.Value := '842';
    oKPI.Icon := '#';
    oKPI.Trend := ktUp;
    oKPI.TrendValue := '+12';
    oKPI.FooterText := 'today';
    oKPI.Theme := utDark;
    oKPI.AddValue(790);
    oKPI.AddValue(815);
    oKPI.AddValue(842);
    // 320 x 160 logical pixels, a 640 x 320 image at scale 2
    oKPI.SaveToPNG('orders.png', 320, 160, 2);
    oKPI.SaveToSVG('orders.svg', 320, 160);
  finally
    oKPI.Free;
  end;
end;
```

Véase también

- Imágenes KPI sin interfaz

- TsgcVCLKPICard
- TsgcFMXKPICard

TsgcVCLKPICard

Unidad: sgcDashboard_KPI_VCL
TsgcVCLKPICard es la tarjeta KPI de VCL, solo para Windows. El control posee un TsgcDashboardKPI, publica sus propiedades y lo dibuja mediante la base de controles VCL de sgcUICore. Un cambio solo programa un nuevo fotograma, por lo que AddValue puede llamarse desde cualquier hilo y la tarjeta se repinta en el hilo principal, como máximo MaxFPS veces por segundo.

Propiedades

Propiedad	Tipo	Descripción
Title	string	Título en la parte superior de la tarjeta.
Value	string	La cifra grande de la tarjeta, ya formateada.
Icon	string	Un glifo de texto corto dibujado en un círculo de acento, por ejemplo '$' o '%'. Si está vacío no se dibuja icono.
Trend	TsgcDashboardKPITrend	ktNone (predeterminado), ktUp, ktDown o ktNeutral. Dibuja la flecha de tendencia, verde para subida y roja para bajada.
TrendValue	string	Texto junto a la flecha de tendencia, por ejemplo '+4.2 %'.
FooterText	string	Texto pequeño en la parte inferior de la tarjeta.
AccentColor	TsgcUIColor	Color $AARRGGBB del círculo del icono y del sparkline. 0 (predeterminado) usa el color de acento del tema.
Theme	TsgcUITheme	utLight (predeterminado) o utDark. Dentro de un tile de dashboard lo establece el tile.
MaxPoints	Integer	Número de valores que conserva el sparkline. Predeterminado 50. El valor más antiguo se descarta cuando llega uno nuevo.
KPI	TsgcDashboardKPI (public, solo lectura)	El modelo de la tarjeta, que también se puede usar para la salida PNG y SVG sin interfaz con el tamaño que se desee.
MaxFPS	Integer (public)	Número máximo de fotogramas por segundo del control, heredado de la base de controles de sgcUICore.
FrameCount	Cardinal (public, solo lectura)	Número de fotogramas renderizados desde que se creó el control.

También se publican los miembros habituales de un control: **Align**, **Anchors**, **Color**, **Constraints**, **Enabled**, **ParentColor**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **Visible** y los eventos de clic, de ratón y de redimensionado.

Métodos

Método	Devuelve	Descripción

AddValue(aValue: Double)		Añade un valor al sparkline. Seguro entre hilos. Los valores NaN e infinitos se ignoran.
ClearValues		Elimina todos los valores del sparkline.
ContentChanged		Programa un nuevo fotograma. Seguro entre hilos. La tarjeta lo llama sola en cada cambio.

Eventos

Evento	Tipo	Descripción
OnClick	TNotifyEvent	Se hizo clic en la tarjeta.
OnResize	TNotifyEvent	La tarjeta se redimensionó.

Ejemplo

Actualizar una tarjeta con una nueva lectura:

```
procedure UpdateCard(aCard: TsgcVCLKPICard; aOld, aNew: Double);
begin
  aCard.Value := FormatFloat('#,##0', aNew);
  // AddValue is thread-safe, the card repaints on the main thread
  aCard.AddValue(aNew);
  if aNew > aOld then
    aCard.Trend := ktUp
  else if aNew < aOld then
    aCard.Trend := ktDown
  else
    aCard.Trend := ktNeutral;
  aCard.TrendValue := FormatFloat('+0.0;-0.0', aNew - aOld);
end;
```

Véase también

- TsgcDashboardKPI
- TsgcFMXKPICard
- Datos en vivo

TsgcFMXKPICard

Unidad: sgcDashboard_KPI_FMX
TsgcFMXKPICard es la tarjeta KPI de FireMonkey, disponible desde Delphi XE6. Tiene los mismos miembros que TsgcVCLKPICard y dibuja el mismo modelo TsgcDashboardKPI.

Propiedades

Propiedad	Tipo	Descripción
Title	string	Título en la parte superior de la tarjeta.
Value	string	La cifra grande de la tarjeta, ya formateada.
Icon	string	Un glifo de texto corto dibujado en un círculo de acento, por ejemplo '$' o '%'. Si está vacío no se dibuja icono.
Trend	TsgcDashboardKPITrend	ktNone (predeterminado), ktUp, ktDown o ktNeutral. Dibuja la flecha de tendencia, verde para subida y roja para bajada.
TrendValue	string	Texto junto a la flecha de tendencia, por ejemplo '+4.2 %'.
FooterText	string	Texto pequeño en la parte inferior de la tarjeta.
AccentColor	TsgcUIColor	Color $AARRGGBB del círculo del icono y del sparkline. 0 (predeterminado) usa el color de acento del tema.
Theme	TsgcUITheme	utLight (predeterminado) o utDark. Dentro de un tile de dashboard lo establece el tile.
MaxPoints	Integer	Número de valores que conserva el sparkline. Predeterminado 50. El valor más antiguo se descarta cuando llega uno nuevo.
KPI	TsgcDashboardKPI (public, solo lectura)	El modelo de la tarjeta, que también se puede usar para la salida PNG y SVG sin interfaz.
MaxFPS	Integer (public)	Número máximo de fotogramas por segundo del control, heredado de la base de controles de sgcUICore.

También se publican los miembros habituales de un control FMX, entre ellos **Align**, **Anchors**, **Margins**, **Opacity**, **RotationAngle**, **Scale**, **Visible**, y los eventos de ratón, de dibujo y de redimensionado.

Métodos

Método	Devuelve	Descripción
AddValue(aValue: Double)		Añade un valor al sparkline. Seguro entre hilos. Los valores NaN e infinitos se ignoran.
ClearValues		Elimina todos los valores del sparkline.

Eventos

Evento	Tipo	Descripción
OnClick	TNotifyEvent	Se hizo clic en la tarjeta.
OnResize	TNotifyEvent	La tarjeta se redimensionó.

Ejemplo

```
procedure ConfigureCard(aCard: TsgcFMXKPICard);
begin
  aCard.Title := 'CPU';
  aCard.Icon := '%';
  aCard.FooterText := 'last 60 samples';
  // $AARRGGBB, 0 uses the accent colour of the theme
  aCard.AccentColor := $FF0078D4;
  aCard.MaxPoints := 60;
  aCard.ClearValues;
end;
```

Véase también

- TsgcDashboardKPI
- TsgcVCLKPICard
- Inicio rápido FMX

TsgcDashboardSource

Unidad: sgcDashboardWS_Source
TsgcDashboardSource es la clase base de los orígenes del puente de sgcWebSockets que alimentan una tarjeta KPI: TsgcDashboardSource_WebSocket, TsgcDashboardSource_MQTT y TsgcDashboardSource_SSE. Cada mensaje que recibe un descendiente se lee como JSON: el número en **ValuePath** se añade al sparkline de la tarjeta y pasa a ser su Value, con el formato de **ValueFormat**. El Trend de la tarjeta se deriva del valor anterior del mismo origen: mayor es ktUp, menor es ktDown e igual es ktNeutral. El puente no forma parte de la edición Community. Necesita los paquetes sgcDashboardWS y sgcWebSockets.
El manejo de hilos se explica en Datos en vivo.

Propiedades

Propiedad	Tipo	Descripción
Active	Boolean	Cuando es False el origen ignora todos los mensajes y restaura el manejador anterior del cliente. Predeterminado True.
Card	TComponent	La tarjeta KPI que se alimenta: un TsgcVCLKPICard o un TsgcFMXKPICard, cualquier componente que implemente IsgcDashboardKPIHost. Se rechazan otros componentes.
ValuePath	string	Ruta JSON del número en cada mensaje, por ejemplo 'price' o 'data.items[0].value'. Se acepta un número o una cadena numérica, con '.' como separador decimal sea cual sea la configuración regional. Vacío lee todo el mensaje como el número.
ValueFormat	string	Cadena de formato para el texto Value de la tarjeta, pasada a Format con el número. Predeterminado '%.2f'. La salida usa el separador decimal de la configuración regional de la aplicación.
TitlePath	string	Si se establece, copia ese valor JSON en el Title de la tarjeta.
TrendValuePath	string	Si se establece, copia ese valor JSON en el TrendValue de la tarjeta.

Eventos

Evento	Tipo	Descripción
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Se leyó un valor y se actualizó la tarjeta. Hilo principal en Delphi 2010 y posteriores.

Ejemplo

```
procedure TFeedHandler.DoValue(Sender: TObject; aValue: Double);
begin
  // main thread on Delphi 2010 and later, after the card was updated
  if aValue > 100 then
    WriteLn('price above 100: ', FormatFloat('0.00', aValue));
end;
```

Véase también

- Datos en vivo
- TsgcVCLKPICard

TsgcDashboardSource_WebSocket

Unidad: sgcDashboardWS_Source_WebSocket
TsgcDashboardSource_WebSocket alimenta una tarjeta KPI con los mensajes de texto de un TsgcWebSocketClient. Mientras el origen está Active y Client está asignado, solo en tiempo de ejecución, el origen instala su propio manejador OnMessage en el cliente y conserva el manejador que había antes: cada mensaje de texto se lee primero en la tarjeta y después se pasa a ese manejador anterior, de modo que el manejador de la aplicación sigue disparándose. Al establecer Active en False, vaciar Client o destruir el origen se restaura el manejador anterior, salvo que la aplicación lo haya reemplazado mientras tanto. El puente no forma parte de la edición Community. Necesita los paquetes sgcDashboardWS y sgcWebSockets.

Propiedades

Propiedad	Tipo	Descripción
Client	TsgcWebSocketClient	El cliente WebSocket cuyos mensajes de texto se leen.
Active	Boolean	Cuando es False el origen ignora todos los mensajes y restaura el manejador anterior del cliente. Predeterminado True.
Card	TComponent	La tarjeta KPI que se alimenta: un TsgcVCLKPICard o un TsgcFMXKPICard, cualquier componente que implemente IsgcDashboardKPIHost. Se rechazan otros componentes.
ValuePath	string	Ruta JSON del número en cada mensaje, por ejemplo 'price' o 'data.items[0].value'. Se acepta un número o una cadena numérica, con '.' como separador decimal sea cual sea la configuración regional. Vacío lee todo el mensaje como el número.
ValueFormat	string	Cadena de formato para el texto Value de la tarjeta, pasada a Format con el número. Predeterminado '%.2f'. La salida usa el separador decimal de la configuración regional de la aplicación.
TitlePath	string	Si se establece, copia ese valor JSON en el Title de la tarjeta.
TrendValuePath	string	Si se establece, copia ese valor JSON en el TrendValue de la tarjeta.

Métodos

Método	Devuelve	Descripción
ProcessMessage(const aText: string)	Boolean	Lee un mensaje como si viniera de Client. False cuando no es JSON válido o ValuePath no es un número. Útil para pruebas y para mensajes que llegan por otra vía.

Eventos

Evento	Tipo	Descripción
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Se leyó un valor y se actualizó la tarjeta. Hilo principal en Delphi 2010 y posteriores.

Ejemplo

```
procedure BindWebSocket;
begin
  WSClient := TsgcWebSocketClient.Create(nil);
  WSClient.URL := 'ws://127.0.0.1:5416';

  // messages look like {"price": 101.25, "change": "+0.4 %"}
  SourceWS := TsgcDashboardSource_WebSocket.Create(nil);
  SourceWS.Client := WSClient;
  SourceWS.Card := CardPrice;
  SourceWS.ValuePath := 'price';
  SourceWS.TrendValuePath := 'change';
  SourceWS.ValueFormat := '%.2f USD';

  WSClient.Active := True;
end;
```

Véase también

- TsgcDashboardSource
- TsgcDashboardSource_MQTT
- Datos en vivo

TsgcDashboardSource_MQTT

Unidad: sgcDashboardWS_Source_MQTT
TsgcDashboardSource_MQTT alimenta una tarjeta KPI con las publicaciones de un TsgcWSPClient_MQTT. Mientras el origen está Active y Client está asignado, solo en tiempo de ejecución, el origen instala su propio manejador OnMQTTPublish en el cliente y conserva el manejador que había antes: cada publicación se lee primero en la tarjeta y después se pasa a ese manejador anterior, de modo que el manejador de la aplicación sigue disparándose. Al establecer Active en False, vaciar Client o destruir el origen se restaura el manejador anterior, salvo que la aplicación lo haya reemplazado mientras tanto. Una aplicación que asigna Client.OnMQTTPublish después de conectar el origen debe asignarlo antes de establecer Client, o alternar Active, para conservar ambos.
Topic selecciona las publicaciones que se leen, con los comodines MQTT + (un nivel) y # (este nivel y todos los inferiores). Un Topic vacío lee todas las publicaciones. El origen no se suscribe, la aplicación se suscribe a través del cliente MQTT. Un contenido que sea un simple número se lee con un ValuePath vacío. El puente no forma parte de la edición Community. Necesita los paquetes sgcDashboardWS y sgcWebSockets.

Propiedades

Propiedad	Tipo	Descripción
Client	TsgcWSPClient_MQTT	El cliente MQTT cuyas publicaciones se leen.
Topic	string	Filtro de tema MQTT de las publicaciones que se leen. Vacío lee todas las publicaciones.
Active	Boolean	Cuando es False el origen ignora todos los mensajes y restaura el manejador anterior del cliente. Predeterminado True.
Card	TComponent	La tarjeta KPI que se alimenta: un TsgcVCLKPICard o un TsgcFMXKPICard, cualquier componente que implemente IsgcDashboardKPIHost. Se rechazan otros componentes.
ValuePath	string	Ruta JSON del número en cada mensaje, por ejemplo 'price' o 'data.items[0].value'. Se acepta un número o una cadena numérica, con '.' como separador decimal sea cual sea la configuración regional. Vacío lee todo el mensaje como el número.
ValueFormat	string	Cadena de formato para el texto Value de la tarjeta, pasada a Format con el número. Predeterminado '%.2f'. La salida usa el separador decimal de la configuración regional de la aplicación.
TitlePath	string	Si se establece, copia ese valor JSON en el Title de la tarjeta.
TrendValuePath	string	Si se establece, copia ese valor JSON en el TrendValue de la tarjeta.

Métodos

Método	Devuelve	Descripción
ProcessPublish(const aTopic, aText: string)	Boolean	Lee una publicación como si viniera de Client. False cuando el tema no coincide con Topic o no se puede leer el contenido.

Eventos

Evento	Tipo	Descripción
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Se leyó un valor y se actualizó la tarjeta. Hilo principal en Delphi 2010 y posteriores.

Ejemplo

```
procedure BindMQTT;
begin
  MQTT := TsgcWSPClient_MQTT.Create(nil);
  MQTT.Client := WSClient;

  // payloads look like {"sensor": "hall", "value": 21.5}
  SourceMQTT := TsgcDashboardSource_MQTT.Create(nil);
  SourceMQTT.Client := MQTT;
  SourceMQTT.Card := CardTemp;
  SourceMQTT.Topic := 'sensors/+/temperature';
  SourceMQTT.ValuePath := 'value';
  SourceMQTT.ValueFormat := '%.1f C';
  // the source does not subscribe: call MQTT.Subscribe in OnMQTTConnect
end;
```

Véase también

- TsgcDashboardSource
- TsgcDashboardSource_WebSocket
- Datos en vivo

TsgcDashboardSource_SSE

Unidad: sgcDashboardWS_Source_SSE
TsgcDashboardSource_SSE alimenta una tarjeta KPI con los eventos enviados por el servidor de un TsgcHTTP1Client. La aplicación inicia el flujo con **Client.GetSSE(URL)**, que bloquea hasta que termina el flujo, por lo que normalmente se llama desde un hilo de trabajo. Mientras el origen está Active y Client está asignado, solo en tiempo de ejecución, el origen instala su propio manejador OnSSEMessage en el cliente y conserva el manejador que había antes: cada fragmento se lee primero en la tarjeta y después se pasa a ese manejador anterior, de modo que el manejador de la aplicación sigue disparándose. Al establecer Active en False, vaciar Client o destruir el origen se restaura el manejador anterior, salvo que la aplicación lo haya reemplazado mientras tanto. El manejador de la aplicación todavía puede cancelar el flujo.
OnSSEMessage entrega los bytes en bruto del flujo a medida que llegan, por lo que un fragmento puede contener varios eventos o parte de uno. El origen almacena los fragmentos en un búfer, los divide en eventos en cada línea en blanco (finales de línea LF, CRLF o CR), une con LF las líneas data: de un evento, ignora los comentarios y los demás campos, y lee los datos de cada evento mediante ValuePath. Un evento sin datos se ignora. El puente no forma parte de la edición Community. Necesita los paquetes sgcDashboardWS y sgcWebSockets.

Propiedades

Propiedad	Tipo	Descripción
Client	TsgcHTTP1Client	El cliente HTTP cuyos eventos enviados por el servidor se leen.
Active	Boolean	Cuando es False el origen ignora todos los mensajes y restaura el manejador anterior del cliente. Predeterminado True.
Card	TComponent	La tarjeta KPI que se alimenta: un TsgcVCLKPICard o un TsgcFMXKPICard, cualquier componente que implemente IsgcDashboardKPIHost. Se rechazan otros componentes.
ValuePath	string	Ruta JSON del número en cada mensaje, por ejemplo 'price' o 'data.items[0].value'. Se acepta un número o una cadena numérica, con '.' como separador decimal sea cual sea la configuración regional. Vacío lee todo el mensaje como el número.
ValueFormat	string	Cadena de formato para el texto Value de la tarjeta, pasada a Format con el número. Predeterminado '%.2f'. La salida usa el separador decimal de la configuración regional de la aplicación.
TitlePath	string	Si se establece, copia ese valor JSON en el Title de la tarjeta.
TrendValuePath	string	Si se establece, copia ese valor JSON en el TrendValue de la tarjeta.

Métodos

Método	Devuelve	Descripción
ProcessChunk(const aText: string)	Integer	Lee los eventos completos de un fragmento del flujo como si vinieran de Client. Un evento incompleto espera al siguiente fragmento. Devuelve el número de eventos cuyos datos dieron un valor.

ResetStream		Descarta un evento incompleto almacenado en el búfer, por ejemplo antes de iniciar un flujo nuevo.

Eventos

Evento	Tipo	Descripción
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Se leyó un valor y se actualizó la tarjeta. Hilo principal en Delphi 2010 y posteriores.

Ejemplo

```
procedure BindSSE;
begin
  HTTP := TsgcHTTP1Client.Create(nil);

  // events look like  data: {"cpu": {"load": 37.5}}
  SourceSSE := TsgcDashboardSource_SSE.Create(nil);
  SourceSSE.Client := HTTP;
  SourceSSE.Card := CardLoad;
  SourceSSE.ValuePath := 'cpu.load';
  SourceSSE.ValueFormat := '%.0f %%';
  // GetSSE blocks until the stream ends, call it from a worker thread:
  // HTTP.GetSSE('https://www.example.com/stream');
end;
```

Véase también

- TsgcDashboardSource
- TsgcDashboardSource_WebSocket
- Datos en vivo

Disposición

El dashboard coloca sus tiles en una cuadrícula. La cuadrícula y las celdas son las mismas en VCL y FMX, y las calcula el motor de disposición independiente del framework TsgcDashboardLayout (unidad sgcDashboard_Layout), expuesto mediante la propiedad **Layout** del dashboard.

Métricas de la cuadrícula

Propiedad	Predeterminado	Significado
Columns	12	Número de columnas. El ancho de columna es el ancho del dashboard, menos Padding a ambos lados y los huecos, dividido entre Columns.
RowHeight	120	Altura de una fila en píxeles independientes del dispositivo. Un dashboard tiene tantas filas como necesiten sus tiles y se desplaza verticalmente.
Gap	12	Espacio entre dos tiles, en horizontal y en vertical.
Padding	12	Espacio entre el borde del dashboard y la cuadrícula.
AutoFlow	True	Coloca en el primer hueco libre los tiles cuyo Col o Row es -1.

Todos los tamaños son píxeles independientes del dispositivo. El dashboard VCL los escala por el PPI del monitor dividido entre 96, el dashboard FMX por la escala del formulario.

Celdas de los tiles

Cada tile solicita una celda con **Col**, **Row**, **ColSpan** y **RowSpan**. Col y Row están basados en cero. Un tile nuevo tiene Col y Row en -1 y una extensión de 3 columnas por 2 filas. Antes de colocar un tile, el motor limita su celda: ColSpan de 1 hasta Columns, RowSpan como mínimo 1, y Col para que el tile no sobrepase la última columna.

Regla de colocación y colisión

- **Pasada 1, celdas explícitas.** Los tiles con Col y Row establecidos se colocan en orden de control. Cuando la celda se solapa con un tile ya colocado, el tile se empuja hacia abajo una fila cada vez hasta que su celda queda libre. Sus columnas nunca cambian.
- **Pasada 2, colocación automática.** Mientras AutoFlow es True, los tiles cuyo Col o Row es -1 ocupan el primer hueco libre, recorriendo las filas desde arriba y las columnas desde la izquierda. Una coordenada dada restringe la búsqueda: un tile con Col 6 y Row -1 va a la primera fila libre en la columna 6.
- Cuando AutoFlow es False, un Col o Row de -1 se interpreta como 0 y el tile se coloca en la pasada 1.
- Un tile oculto (Visible False) no ocupa ninguna celda.

La celda solicitada permanece en Col, Row, ColSpan y RowSpan. La celda colocada, tras la limitación y la resolución de colisiones, está en **Layout.PlacedCell[Index]** y su rectángulo en **Layout.Rect[Index]**.

Ejemplo

```
constructor TfrmMain.Create(aOwner: TComponent);
var
  oTile: TsgcVCLDashboardTile;
  oCard: TsgcVCLKPICard;
  oMemo: TMemo;
begin
  inherited CreateNew(aOwner);
```

```
  Caption := 'sgcDashboard VCL Quick Start';
  Width := 900;
  Height := 560;

  // the dashboard: a 12 column grid that fills the form
  FDashboard := TsgcVCLDashboard.Create(Self);
  FDashboard.Parent := Self;
  FDashboard.Align := alClient;
  FDashboard.Columns := 12;
  FDashboard.RowHeight := 110;

  // a KPI tile at column 0, row 0, 3 columns wide and 2 rows high
  oTile := TsgcVCLDashboardTile.Create(Self);
  oTile.Name := 'tileRevenue';
  oTile.Parent := FDashboard;
  oTile.Col := 0;
  oTile.Row := 0;
  oTile.ColSpan := 3;
  oTile.RowSpan := 2;
  oTile.ShowHeader := False;

  oCard := TsgcVCLKPICard.Create(Self);
  oCard.Parent := oTile;
  oCard.Align := alClient;
  oCard.Title := 'Revenue';
  oCard.Value := '12,480 USD';
  oCard.Icon := '$';
  oCard.Trend := ktUp;
  oCard.TrendValue := '+4.2 %';
  oCard.FooterText := 'vs last week';
  oCard.AddValue(11200);
  oCard.AddValue(11900);
  oCard.AddValue(12480);

  // a log tile, Col and Row stay -1: auto-flow puts it in the first free slot
  oTile := TsgcVCLDashboardTile.Create(Self);
  oTile.Name := 'tileEvents';
  oTile.Parent := FDashboard;
  oTile.Title := 'Events';
  oTile.ColSpan := 6;
  oTile.RowSpan := 3;

  oMemo := TMemo.Create(Self);
  oMemo.Parent := oTile;
  oMemo.Align := alClient;
  oMemo.Lines.Add('Service started');

  // one property themes the dashboard, the tiles and the hosted controls
  FDashboard.Theme := utDark;
end;
```

Véase también

- TsgcVCLDashboard
- TsgcVCLDashboardTile
- Persistencia

Temas

La propiedad **Theme** del dashboard, de tipo TsgcUITheme de sgcUICore_Theme, selecciona **utLight** (predeterminado), **utDark** o **utCustom**. El dashboard dibuja su fondo con el tema y lo sigue cada tile cuyo **ParentTheme** es True. Un tile puede establecer su propio **Theme**, lo que desactiva ParentTheme. El tema se aplica cuando cambia y cuando se inserta un control en un tile.

Qué establece el aplicador de temas en los controles alojados

Un tile aplica su tema a cada control que aloja mediante las propiedades publicadas que expone el control, sin dependencia en tiempo de compilación del producto alojado. Gana la primera regla que coincide:

1. Se establece un **Theme** publicado y escribible de tipo TsgcUITheme. Esto cubre TsgcVCLKPICard y TsgcFMXKPICard.
2. Si no, se aplica el tema a un objeto **Chart** publicado que tenga ese Theme: **Chart.Theme** de un TsgcVCLChart o TsgcFMXChart de sgcCharts.
3. Si no, un **BackgroundColor** publicado y escribible toma el color de superficie del tema, por ejemplo el visor de sgcPDF.
4. Si no, un **Color** publicado y escribible toma el color de superficie, por ejemplo el mapa de sgcMaps, un TMemo o un TListBox.

Cuando se llega a la regla 3 o 4, también se establece el color de texto del tema para que el texto siga siendo legible sobre una superficie oscura: **Font.Color** en VCL, y **TextSettings.FontColor** o un **FontColor** publicado en FMX. Un objeto **Viewer** publicado de un control alojado recibe el tema con las mismas reglas, un nivel de profundidad.
Con **utCustom** el aplicador no toca ningún control alojado, por lo que la aplicación conserva el control total de sus colores. Llame a **ApplyTheme** en el dashboard para volver a aplicar el tema después de cambiar controles por código.

Límites conocidos

- La vista previa de sgcReports no publica ninguna propiedad de color, por lo que conserva sus propios colores en ambos temas.
- Un control que no publica ninguna de las propiedades anteriores no se modifica.
- Solo reciben el tema los controles hijos directos de un tile, más su objeto Viewer. Los controles anidados dentro de un panel del tile conservan sus colores.

Ejemplo

```
procedure ApplyThemes(aDashboard: TsgcVCLDashboard;
  aTileLight: TsgcVCLDashboardTile);
begin
  // every tile with ParentTheme = True follows the dashboard
  aDashboard.Theme := utDark;
  // this tile keeps its own theme: setting Theme clears ParentTheme
  aTileLight.Theme := utLight;
  // back to the dashboard theme
  aTileLight.ParentTheme := True;
  // pushes the theme again, after controls were changed in code
  aDashboard.ApplyTheme;
end;
```

Véase también

- TsgcVCLDashboardTile
- TsgcVCLKPICard

- Demos

Persistencia

Una build Registered o Trial permite al usuario reorganizar los tiles en tiempo de ejecución y guarda la disposición como JSON. Estos miembros se compilan con SGC_DASHBOARD_PERSIST y no existen en la edición Community.

Arrastre y redimensionado en tiempo de ejecución

- **AllowDrag**: pulsar el botón izquierdo del ratón sobre la cabecera de un tile lo mueve. Un tile con ShowHeader en False no se puede mover.
- **AllowResize**: se dibuja un asa en la esquina inferior derecha de cada tile, y arrastrarla redimensiona el tile.
- Mientras se mueve el ratón, la celda de destino se dibuja debajo de los tiles y el tile se ajusta a la cuadrícula. Escape cancela el arrastre.
- **Locked**: un tile bloqueado nunca se mueve ni se redimensiona. En FMX es la propiedad estándar Locked.
- **OnLayoutChanged** se dispara después de que un arrastre o redimensionado cambie una celda, y después de LoadLayout.

```
procedure SetupDashboard(aDashboard: TsgcVCLDashboard;
  aTileLocked: TsgcVCLDashboardTile; aOnChanged: TNotifyEvent);
begin
  // the user moves a tile by its header and resizes it by its grip
  aDashboard.AllowDrag := True;
  aDashboard.AllowResize := True;
  // this tile never moves
  aTileLocked.Locked := True;
  aDashboard.OnLayoutChanged := aOnChanged;
end;
```

SaveLayout y LoadLayout

SaveLayout devuelve las métricas de la cuadrícula y, para cada tile con Name, su celda y su visibilidad. **LoadLayout** aplica un texto así: los tiles se emparejan por Name, los nombres desconocidos se ignoran y los tiles que faltan en el texto conservan sus celdas. Las variantes de archivo y de flujo usan UTF-8 sin BOM y omiten un BOM al cargar.

```
procedure SaveUserLayout(aDashboard: TsgcVCLDashboard);
begin
  aDashboard.SaveLayoutToFile(ExtractFilePath(ParamStr(0)) + 'layout.json');
end;

procedure LoadUserLayout(aDashboard: TsgcVCLDashboard);
var
  vFile: string;
begin
  vFile := ExtractFilePath(ParamStr(0)) + 'layout.json';
  if FileExists(vFile) then
    aDashboard.LoadLayoutFromFile(vFile);
end;
```

Formato JSON

Esta es la salida de SaveLayout para un dashboard con dos tiles con nombre, formateada aquí para facilitar la lectura. SaveLayout la escribe en una sola línea.

```
{
  "version":1,
  "columns":12,
  "rowHeight":120,
  "gap":12,
```

```
  "padding":12,
  "tiles": [
    {"name":"tileSales","col":0,"row":0,"colSpan":6,"rowSpan":2,"visible":true},
    {"name":"tileLog","col":6,"row":0,"colSpan":6,"rowSpan":3,"visible":true}
  ]
}
```

Clave	Significado
version	Versión del formato, 1.
columns, rowHeight, gap, padding	Las métricas de la cuadrícula del dashboard.
tiles	Un objeto por cada tile con nombre: name, col, row, colSpan, rowSpan y visible.

Véase también

- TsgcVCLDashboard
- Disposición
- Demos
- Ediciones

Datos en vivo

Una tarjeta KPI se alimenta por código con **Value**, **Trend** y **AddValue**, o desde una conexión en vivo mediante el puente sgcDashboardWS. Los orígenes del puente leen un número de cada mensaje JSON de un cliente de sgcWebSockets y lo envían a una tarjeta. El puente no forma parte de la edición Community.

Origen	Cliente	Lee
TsgcDashboardSource_WebSocket	TsgcWebSocketClient	Cada mensaje de texto.
TsgcDashboardSource_MQTT	TsgcWSPClient_MQTT	Cada publicación cuyo tema coincide con Topic.
TsgcDashboardSource_SSE	TsgcHTTP1Client	Los datos de cada evento enviado por el servidor.

Rutas y formato

- **ValuePath** es la ruta JSON del número, por ejemplo price o data.items[0].value. Se acepta un número o una cadena numérica, con . como separador decimal sea cual sea la configuración regional. Un ValuePath vacío lee todo el mensaje como el número.
- **ValueFormat** da formato al número para el texto Value de la tarjeta con Format. El valor predeterminado es %.2f. El texto usa el separador decimal de la configuración regional de la aplicación.
- **TitlePath** y **TrendValuePath**, cuando se establecen, copian ese valor JSON en el Title y el TrendValue de la tarjeta.
- El Trend de la tarjeta sigue el valor anterior del mismo origen: mayor es ktUp, menor es ktDown, igual es ktNeutral.
- Un mensaje que no es JSON válido, o cuyo ValuePath no es un número, se ignora.

Hilos

- El número se añade al sparkline de inmediato, en el hilo en el que llega el mensaje. TsgcDashboardKPI.AddValue es seguro entre hilos.
- En Delphi 2010 y posteriores, las actualizaciones de texto (Value, Title, TrendValue, Trend) se aplican en el hilo principal mediante TThread.Queue, seguidas de OnValue. El cliente puede usar cualquier modo de NotifyEvents.
- Las aplicaciones de consola y los servicios no tienen un bucle de mensajes que ejecute la cola, por lo que deben llamar a CheckSynchronize.
- Delphi 7 a 2009 no tienen TThread.Queue. En ellos la actualización y OnValue se ejecutan en el hilo en el que llegó el mensaje, y un manejador de OnValue debe sincronizarse por sí mismo antes de tocar un control.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP,
  sgcDashboard_KPI_VCL, sgcDashboardWS_Source_WebSocket,
  sgcDashboardWS_Source_MQTT, sgcDashboardWS_Source_SSE;

procedure BindWebSocket;
begin
  WSClient := TsgcWebSocketClient.Create(nil);
  WSClient.URL := 'ws://127.0.0.1:5416';

  // messages look like {"price": 101.25, "change": "+0.4 %"}
```

```
  SourceWS := TsgcDashboardSource_WebSocket.Create(nil);
  SourceWS.Client := WSClient;
  SourceWS.Card := CardPrice;
  SourceWS.ValuePath := 'price';
  SourceWS.TrendValuePath := 'change';
  SourceWS.ValueFormat := '%.2f USD';

  WSClient.Active := True;
end;
```

Véase también

- TsgcDashboardSource
- TsgcVCLKPICard
- Demos

Imágenes KPI sin interfaz

TsgcDashboardKPI renderiza una tarjeta KPI sin ventana y sin VCL ni FMX, de modo que una aplicación de consola, un servicio o un servidor web pueden producir la misma tarjeta como imagen, por ejemplo para un informe, un correo electrónico o una página web. La unidad se compila en todas las plataformas, incluida Linux64.

- **SaveToPNG** escribe un archivo o flujo PNG. aWidth y aHeight son píxeles lógicos y aScale los multiplica: 320 por 160 con escala 2 da una imagen de 640 por 320.
- **SaveToSVG** escribe un archivo o flujo SVG de la misma tarjeta.
- El modelo de la tarjeta de un control está disponible mediante su propiedad **KPI**, por lo que **Card.KPI.SaveToPNG** guarda lo que muestra el control con cualquier tamaño.

Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcDashboard_KPI;

procedure SaveKPIImages;
var
  oKPI: TsgcDashboardKPI;
begin
  oKPI := TsgcDashboardKPI.Create;
  try
    oKPI.Title := 'Orders';
    oKPI.Value := '842';
    oKPI.Icon := '#';
    oKPI.Trend := ktUp;
    oKPI.TrendValue := '+12';
    oKPI.FooterText := 'today';
    oKPI.Theme := utDark;
    oKPI.AddValue(790);
    oKPI.AddValue(815);
    oKPI.AddValue(842);
    // 320 x 160 logical pixels, a 640 x 320 image at scale 2
    oKPI.SaveToPNG('orders.png', 320, 160, 2);
    oKPI.SaveToSVG('orders.svg', 320, 160);
  finally
    oKPI.Free;
  end;
end;
```

Véase también

- TsgcDashboardKPI
- Plataformas

Visión general de .NET

sgcDashboard .NET es un port administrado en C# de la biblioteca Delphi sgcDashboard, no un contenedor (wrapper) que la envuelva. Las clases de C# están en el espacio de nombres **esegece.sgcDashboard** y conservan los nombres de Delphi, por lo que **TsgcDashboardLayout**, **TsgcDashboardCell**, **TsgcDashboardKPI**, **TsgcDashboardKPITrend** e **IsgcDashboardKPIHost** se leen igual en ambos lenguajes, y el dashboard, el tile y la tarjeta KPI de cada host siguen el patrón de nombres de Delphi. Se dibujan con el port .NET de **sgcUICore**, el mismo lienzo, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, de modo que un dashboard se ve igual en WinForms, WPF, Avalonia y en la salida sin interfaz en PNG y SVG.
Un dashboard organiza tiles en una cuadrícula de columnas y filas. Cada tile tiene una celda (una columna, una fila y la extensión de ambas), una cabecera opcional con un título, y aloja cualquier control: una tarjeta KPI, un gráfico de sgcCharts, un visor de sgcPDF, una vista previa de sgcReports o un cuadro de texto común. El tema del dashboard llega a los tiles y a los controles que alojan. En la edición Registered el usuario puede arrastrar y redimensionar los tiles en tiempo de ejecución, la disposición se guarda y se carga como JSON, y el puente de sgcWebSockets alimenta las tarjetas KPI desde una conexión en vivo.

Paquetes

El port se divide en seis paquetes NuGet, de modo que un proyecto solo incorpora el host que utiliza.

PackageId	Contiene	Frameworks de destino
esegece.sgcDashboard	El motor de disposición **TsgcDashboardLayout** con su persistencia JSON, el modelo KPI **TsgcDashboardKPI**, la enumeración de tendencia y la salida SVG sin interfaz. Sin dependencias nativas.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDashboard.Rendering	La salida PNG sin interfaz de una tarjeta KPI (**SaveToPNG**), mediante SkiaSharp.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDashboard.WinForms	**TsgcWinFormsDashboard**, **TsgcWinFormsDashboardTile** y **TsgcWinFormsKPICard**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDashboard.Wpf	**TsgcWpfDashboard**, **TsgcWpfDashboardTile** y **TsgcWpfKPICard**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDashboard.Avalonia	**TsgcAvaloniaDashboard**, **TsgcAvaloniaDashboardTile** y **TsgcAvaloniaKPICard**, sobre Avalonia 11.3.	net8.0, net9.0
esegece.sgcDashboard.sgcWebSockets	El puente de datos en vivo: **TsgcDashboardSource**, **TsgcDashboardSource_WebSocket**, **TsgcDashboardSource_MQTT** y **TsgcDashboardSource_SSE**. Depende del pa-	netstandard2.0,

	quete esegece.sgcWebSockets. Solo en la edición Registered.	net8.0, net9.0

Cada paquete de host depende de esegece.sgcDashboard.Rendering y del paquete sgcUICore del mismo host, por lo que basta con añadir el paquete del host.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcDashboard, .Rendering, .WinForms, .Wpf, .Avalonia y .sgcWebSockets	Todo: la cuadrícula y los tiles, las tarjetas KPI, los temas, la salida sin interfaz, el arrastre y el redimensionado en tiempo de ejecución, **SaveLayout** y **LoadLayout**, **OnLayoutChanged** y el puente de sgcWebSockets.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcDashboard.WinForms.Community. No hay un paquete de puente Community.	La cuadrícula y los tiles, las tarjetas KPI, los temas y la salida sin interfaz. El arrastre y el redimensionado en tiempo de ejecución, la persistencia de la disposición (**SaveLayout**, **LoadLayout** y los **SaveToJSON** y **LoadFromJSON** del motor) y el puente quedan fuera.

Es la misma división que las ediciones de Delphi, véase Ediciones.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	TsgcVCLDashboard, TsgcVCLDashboardTile y TsgcVCLKPICard	Windows, en .NET Framework 4.7.2 y .NET 8 y 9.
WPF	TsgcFMXDashboard, TsgcFMXDashboardTile y TsgcFMXKPICard	Windows, en .NET Framework 4.7.2 y .NET 8 y 9.
Avalonia	TsgcFMXDashboard, TsgcFMXDashboardTile y TsgcFMXKPICard	Windows, Linux y macOS, en .NET 8 y 9.
Sin interfaz	TsgcDashboardKPI y el motor de disposición	Cualquier plataforma de .NET Standard 2.0, incluidos los servidores Linux.

Los tiles de WPF y Avalonia siguen el tile de FireMonkey: alojan un elemento y gestionan el espacio a su alrededor mediante **ContentPadding**. El tile de WinForms sigue el tile de VCL y aloja cualquier número de controles.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF

- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Renderizado sin interfaz en .NET
- Feeds en vivo en .NET
- Demos de .NET

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, los paquetes core y de renderizado llegan con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcDashboard.WinForms
dotnet add package esegece.sgcDashboard.Wpf
dotnet add package esegece.sgcDashboard.Avalonia
```

Añada el puente para alimentar tarjetas KPI desde sgcWebSockets. Incorpora el paquete esegece.sgcWebSockets:

```
dotnet add package esegece.sgcDashboard.sgcWebSockets
```

Edición Community

```
dotnet add package esegece.sgcDashboard.WinForms.Community
dotnet add package esegece.sgcDashboard.Wpf.Community
dotnet add package esegece.sgcDashboard.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcDashboard.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los Community, nunca ambos: contienen los mismos tipos en el mismo espacio de nombres. El código que usa **AllowDrag**, **AllowResize**, **SaveLayout**, **LoadLayout**, **OnLayoutChanged** o la propiedad **Locked** de un tile no compila con los paquetes Community; protéjalo con `#if ! SGC_COMMUNITY` cuando un proyecto compile ambas ediciones.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto de .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto de .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:dash="clr-namespace:esegece.sgcDashboard;assembly=esegece.sgcDashboard.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Asigne el espacio de nombres en XAML con `xmlns:dash="using:esegece.sgcDashboard"`.
- Sin interfaz: esegece.sgcDashboard por sí solo basta para el motor de disposición y para SVG. Añada esegece.sgcDashboard.Rendering para PNG, que incorpora SkiaSharp y sus recursos nativos para Windows y Linux.
- Puente: esegece.sgcDashboard.sgcWebSockets tiene como destino netstandard2.0, net8.0 y net9.0, por lo que funciona con todos los hosts, incluido .NET Framework 4.7.2.

En C# todo host necesita `using esegece.sgcDashboard;`, y `using esegece.sgcUICore;` para **TsgcUITheme** y los demás tipos de los temas. El puente necesita además `using esegece.sgcWebSockets;`.

Véase también

- Visión general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia

- Feeds en vivo en .NET
- Demos de .NET

Inicio rápido WinForms

Este ejemplo crea un formulario WinForms con un dashboard de tres tiles: dos tarjetas KPI y un registro de eventos en un cuadro de texto común. Un temporizador alimenta la tarjeta de ingresos, y una casilla de verificación cambia el dashboard entre el tema claro y el oscuro.

Pasos

- Cree un proyecto WinForms `net8.0-windows` y añada el paquete **esegece.sgcDashboard.WinForms**, véase Instalación desde NuGet.
- Cree un **TsgcWinFormsDashboard**, acóplelo (dock) y establezca **Columns** y **RowHeight**. La cuadrícula siempre se ajusta al ancho del dashboard y se desplaza en vertical.
- Para cada tile cree un **TsgcWinFormsDashboardTile**, asígnele un **Name**, un **Title** y su celda (**Col**, **Row**, **ColSpan** y **RowSpan**), añada el contenido a los **Controls** del tile y el tile a los **Controls** del dashboard. Un control acoplado llena el área situada bajo la cabecera.
- Establezca **Value**, **Trend**, **TrendValue** y **FooterText** de un **TsgcWinFormsKPICard** y llame a **AddValue** para añadir un punto a su sparkline.
- Establezca **Theme** del dashboard: los tiles, las tarjetas y el cuadro de texto lo siguen.

Ejemplo

AllowDrag y **AllowResize** permiten al usuario mover un tile por su cabecera y redimensionarlo desde el tirador inferior derecho, ajustado a la cuadrícula. Forman parte de la edición Registered; elimine las dos líneas para los paquetes Community.

Guardar y restaurar la disposición

SaveLayout devuelve las medidas de la cuadrícula y la celda y la visibilidad de cada tile con nombre como JSON, y **LoadLayout** vuelve a aplicar ese texto. **SaveLayoutToFile** y **LoadLayoutFromFile** hacen lo mismo con un archivo, **SaveLayoutToStream** y **LoadLayoutFromStream** con un flujo UTF-8. **OnLayoutChanged** se dispara después de que un arrastre cambie una celda y después de una carga. La persistencia de la disposición forma parte de la edición Registered.
El programa escribe:

```
{"version":1,"columns":12,"rowHeight":120,"gap":12,"padding":12,"tiles":[{"name":"tileSales","col":0,"row":0,"colSpan":6,"r
tileSales at 0,0 size 6x2
tileOrders at 6,0 size 6x2
OnLayoutChanged: 2, round trip: ok
```

El JSON es el texto que escribe el **SaveLayout** de Delphi, por lo que una disposición guardada por una aplicación Delphi se carga en una aplicación .NET y viceversa. Los tiles se asocian por **Name** sin distinguir mayúsculas de minúsculas; un tile sin nombre no se guarda.

Véase también

- Instalación desde NuGet
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Persistencia
- TsgcVCLDashboard
- Demos de .NET

Inicio rápido WPF

El mismo dashboard en WPF. Los tiles y lo que alojan se declaran en XAML: los tiles son el contenido del dashboard, y cada tile toma un elemento como contenido.

Pasos

- Cree un proyecto WPF `net8.0-windows` y añada el paquete **esegece.sgcDashboard.Wpf**, véase Instalación desde NuGet.
- Asigne el espacio de nombres en la ventana con `xmlns:dash="clr-namespace:esegece.sgcDashboard;assembly=esegece.sgcDashboard.Wpf"`.
- Declare un **TsgcWpfDashboard** y, dentro de él, un **TsgcWpfDashboardTile** por tile con su **Title**, **Col**, **Row**, **ColSpan** y **RowSpan**. El contenido de un tile es un **TsgcWpfKPICard** o cualquier otro elemento.
- Dé a los tiles un `x:Name`: es el nombre que guarda **SaveLayout**.
- Alimente las tarjetas y cambie **Theme** desde el code-behind.

Ejemplo

MainWindow.xaml:
MainWindow.xaml.cs:
App.xaml es el de un proyecto WPF nuevo, con `StartupUri="MainWindow.xaml"`. **AllowDrag** y **AllowResize** forman parte de la edición Registered; elimine los dos atributos para los paquetes Community.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- TsgcFMXDashboard
- Demos de .NET

Inicio rápido Avalonia

El mismo dashboard en Avalonia, en Windows, Linux y macOS. Como en WPF, los tiles y su contenido se declaran en XAML.

Pasos

- Cree un proyecto de escritorio Avalonia `net8.0` con **Avalonia.Desktop** y **Avalonia.Themes.Fluent** 11.3, y añada el paquete **esegece.sgcDashboard.Avalonia**, véase Instalación desde NuGet.
- Asigne el espacio de nombres en la ventana con `xmlns:dash="using:esegece.sgcDashboard"`.
- Declare un **TsgcAvaloniaDashboard** y, dentro de él, un **TsgcAvaloniaDashboardTile** por tile. El contenido de un tile es un **TsgcAvaloniaKPICard** o cualquier otro control.
- Alimente las tarjetas y cambie **Theme** desde el code-behind.

Ejemplo

MainWindow.axaml:
MainWindow.axaml.cs:
Program.cs:
El tile gestiona su propio **Padding** a partir de **ContentPadding** y **ShowHeader**, por lo que un **Padding** establecido en XAML se reemplaza. En una pantalla táctil, una pulsación en la cabecera de un tile arrastra el tile y una pulsación en cualquier otro lugar desplaza la cuadrícula, véase Diferencias de .NET respecto a Delphi.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- TsgcFMXDashboard
- Demos de .NET

Clases de .NET

Todas las clases están en el espacio de nombres **esegece.sgcDashboard**. Las clases de host son controles delgados sobre el motor y el modelo independientes del framework, que comparten los tres hosts y la salida sin interfaz.

Clases de host

Host	Clase	Clase base	Delphi
WinForms	TsgcWinFormsDashboard	ScrollableControl	TsgcVCLDashboard
WinForms	TsgcWinFormsDashboardTile	TsgcUIWinFormsControl	TsgcVCLDashboardTile
WinForms	TsgcWinFormsKPICard	TsgcUIWinFormsControl	TsgcVCLKPICard
WPF	TsgcWpfDashboard	FrameworkElement	TsgcFMXDashboard
WPF	TsgcWpfDashboardTile	TsgcUIWpfControl	TsgcFMXDashboardTile
WPF	TsgcWpfKPICard	TsgcUIWpfControl	TsgcFMXKPICard
Avalonia	TsgcAvaloniaDashboard	TsgcUIAvaloniaControl	TsgcFMXDashboard
Avalonia	TsgcAvaloniaDashboardTile	TsgcUIAvaloniaControl	TsgcFMXDashboardTile
Avalonia	TsgcAvaloniaKPICard	TsgcUIAvaloniaControl	TsgcFMXKPICard

Tipos independientes del framework

Tipo	Descripción
TsgcDashboardLayout	El motor de la cuadrícula: **Columns**, **RowHeight**, **Gap**, **Padding** y **AutoFlow**, celdas de los tiles a rectángulos (**Arrange**, **Rect**, **ContentHeight**), flujo automático, desplazamiento hacia abajo en las colisiones, el cálculo de ajuste del arrastre y del redimensionado (**SnapMove**, **SnapResize**) y el JSON de la disposición (**SaveToJSON**, **LoadFromJSON**, edición Registered).
TsgcDashboardCell	Una struct con **Col**, **Row**, **ColSpan** y **RowSpan**. `sgcDashboard_Layout.sgcDashboardCell` crea una.
TsgcDashboardLayoutTile	El nombre, la celda y la visibilidad de un tile en el JSON de la disposición. Edición Registered.
TsgcDashboardKPI	El modelo KPI y el renderizador, véase TsgcDashboardKPI.
TsgcDashboardKPITrend	`ktNone` , `ktUp` , `ktDown` o `ktNeutral` .

IsgcDashboardKPIHost	La implementan los tres controles de tarjeta KPI; el puente acepta cualquier tarjeta que la implemente.

Dashboard

Los mismos miembros en **TsgcWinFormsDashboard**, **TsgcWpfDashboard** y **TsgcAvaloniaDashboard**.

Miembro	Tipo	Descripción
Columns	int	Número de columnas de la cuadrícula, 12 por defecto.
RowHeight	int	Altura de una fila de la cuadrícula en píxeles independientes del dispositivo, 120 por defecto.
Gap	int	Espacio entre los tiles, 12 por defecto.
Padding	int	Espacio alrededor de la cuadrícula, 12 por defecto. Oculta `Control.Padding` en WinForms.
AutoFlow	bool	Coloca los tiles sin celda (Col o Row -1) en las primeras celdas libres. True por defecto.
Theme	TsgcUITheme	Tema del dashboard, de sus tiles y de los controles que alojan.
Layout	TsgcDashboardLayout	El motor de la cuadrícula, de solo lectura. Oculta el evento `Control.Layout` en WinForms.
Tiles	colección	WPF y Avalonia: los tiles, la propiedad de contenido de XAML.
VerticalOffset	double	Avalonia: la posición de desplazamiento de la cuadrícula.
AllowDrag	bool	Mueve un tile por su cabecera en tiempo de ejecución. Edición Registered.
AllowResize	bool	Redimensiona un tile desde su tirador inferior derecho en tiempo de ejecución. Edición Registered.
OnLayoutChanged	TsgcUINotifyEvent	Se dispara después de que un arrastre cambie una celda y después de una carga. Edición Registered.
SaveLayout, LoadLayout	método	La disposición como una cadena JSON. Edición Registered.
SaveLayoutToStream, LoadLayoutFromStream	método	La disposición como un flujo UTF-8. Edición Registered.
SaveLayoutToFile, LoadLayoutFromFile	método	La disposición como un archivo. Edición Registered.
Arrange	método	Vuelve a organizar los tiles.
ApplyTheme	método	Vuelve a aplicar el tema a los tiles y a su contenido, por ejemplo después de que el contenido haya cambiado sus propios colores.

Tile

Los mismos miembros en **TsgcWinFormsDashboardTile**, **TsgcWpfDashboardTile** y **TsgcAvaloniaDashboard-Tile**.

Miembro	Tipo	Descripción
Dashboard	dashboard	El dashboard que contiene el tile, o null.
Title	string	Texto de la cabecera.
ShowHeader	bool	Muestra la banda de cabecera, true por defecto.
Col, Row	int	La celda solicitada, -1 por defecto para que el flujo automático coloque el tile.
ColSpan, RowSpan	int	Tamaño del tile en columnas y filas, 3 y 2 por defecto.
GetCell, SetCell	método	Los cuatro valores como un único **TsgcDashboardCell**.
ParentTheme	bool	Toma el tema del dashboard, true por defecto.
Theme	TsgcUIT-heme	Tema del tile cuando **ParentTheme** es false.
Locked	bool	El tile nunca inicia un arrastre ni un redimensionado. Edición Registered.
Content	elemento	WPF y Avalonia: el elemento alojado, la propiedad de contenido de XAML. En WinForms el tile aloja los controles de su colección **Controls**.
ContentPadding	int	WPF y Avalonia: espacio alrededor del contenido, 8 por defecto; la altura de la cabecera se suma arriba mientras **ShowHeader** es true.

Tarjeta KPI

TsgcWinFormsKPICard, **TsgcWpfKPICard** y **TsgcAvaloniaKPICard** envuelven un TsgcDashboardKPI, disponible en la propiedad **KPI**, y repiten sus miembros: **Title**, **Value**, **Icon**, **Trend**, **TrendValue**, **FooterText**, **AccentColor**, **Theme**, **MaxPoints** y **AddValue**.

Véase también

- Visión general de .NET
- Diferencias de .NET respecto a Delphi
- Renderizado sin interfaz en .NET
- Feeds en vivo en .NET
- Disposición
- Temas

Diferencias de .NET respecto a Delphi

El port .NET conserva los nombres, los valores por defecto y las reglas de disposición de Delphi. Las diferencias siguientes provienen de los frameworks de host.

Los tiles son contenedores hijos

Host	Cómo aloja un tile su contenido
Win-Forms	Un tile es un contenedor `Control`, como el tile de VCL: añada cualquier número de controles a sus **Controls**. Se colocan bajo la cabecera con un margen de 8 píxeles, por lo que un control con `Dock = DockStyle.Fill` llena el área de contenido. El tile lleva el ParentControlDesigner, de modo que se puede soltar un control en él en el diseñador.
WPF	Un tile toma un elemento mediante **Content**, la propiedad de contenido de XAML (`[ContentProperty("Content")]`), como el tile de FireMonkey. El dashboard es el padre lógico de sus tiles, y **Tiles** es su propiedad de contenido, de modo que los tiles se declaran como hijos del elemento dashboard.
Avalo-nia	El mismo modelo que WPF, mediante el atributo `[Content]` en **Content** del tile y en **Tiles** del dashboard. El dashboard es el padre lógico y visual de cada tile, y el tile lo es de su contenido.

En WPF y Avalonia el tile calcula su **Padding** a partir de **ContentPadding** y **ShowHeader**, por lo que un **Padding** establecido en XAML se reemplaza. Un tile oculto: en WPF un tile `Hidden` conserva su celda y un tile `Collapsed` sale de la cuadrícula; en Avalonia un tile con `IsVisible` false sale de la cuadrícula.

Propagación del tema a los controles alojados

Cuando cambia el **Theme** del dashboard, cuando un tile entra en un dashboard, cuando se añade contenido a un tile y en **ApplyTheme**, los controles alojados toman el tema del tile mediante los helpers de sgcUICore `sgcUIApplyThemeWinForms`, `sgcUIApplyThemeWpf` y `sgcUIApplyThemeAvalonia`, los equivalentes de los `sgcUIApplyThemeVCL` y `sgcUIApplyThemeFMX` de Delphi. Así, un gráfico de sgcCharts, un visor de sgcPDF o una vista previa de sgcReports en un tile cambia entre claro y oscuro junto con el dashboard. `utCustom` deja intactos los controles alojados.

El JSON de la disposición vive en el motor

SaveToJSON y **LoadFromJSON** de **TsgcDashboardLayout** escriben y leen el texto de la disposición; los métodos **SaveLayout** y **LoadLayout** de los tres hosts solo recogen sus tiles y los llaman. El texto es el mismo que escribe el dashboard de Delphi, por lo que las disposiciones se mueven libremente entre Delphi, WinForms, WPF y Avalonia. La clave de un tile es `Control.Name` en WinForms, `FrameworkElement.Name` (`x:Name`) en WPF y `StyledElement.Name` en Avalonia. Véase Renderizado sin interfaz en .NET para un ejemplo sin host.

Padding y Layout en WinForms

TsgcWinFormsDashboard declara dos miembros con `new`: su **Padding** es el espacio alrededor de la cuadrícula, un `int`, y oculta `Control.Padding`, y su propiedad **Layout** devuelve el motor de la cuadrícula y oculta el evento `Control.Layout`. Convierta el dashboard a `Control` para acceder a los miembros ocultos.

Theme en Avalonia

Todo control de Avalonia ya tiene una propiedad **Theme**, el `ControlTheme` de `StyledElement`. El dashboard, el tile y la tarjeta KPI declaran su propio **Theme** de tipo **TsgcUITheme** con `new`. A través de una referencia del tipo sgc es el tema sgc; a través de una referencia tipada como `Control` o `StyledElement` es el tema de control de Avalonia.

Arrastre y desplazamiento táctil en Avalonia

Un contacto táctil que se mueve más de 8 unidades en vertical desplaza la cuadrícula. Con **AllowDrag** o **AllowResize**, una pulsación táctil en la cabecera o en el tirador de un tile inicia de inmediato el arrastre del tile y nunca desplaza la cuadrícula, como en FireMonkey; una pulsación táctil en cualquier otro lugar sigue desplazando la cuadrícula. Un arrastre con el ratón nunca desplaza la cuadrícula. En WPF el tacto y el lápiz se promueven a entrada de ratón, y WinForms solo usa el ratón.

El contorno del arrastre

Mientras se arrastra o se redimensiona un tile, la celda de destino se marca con el color de acento y un relleno del 25 por ciento. WinForms la dibuja bajo los tiles, como el dashboard de VCL. WPF la dibuja en una capa de adorner y Avalonia en un visual de superposición, ambos sobre los tiles, mientras que el marcador de posición de FireMonkey queda debajo de ellos. **Escape** cancela el arrastre en todos los hosts, y en WPF y Avalonia también lo cancela una captura perdida.

Desplazamiento

La cuadrícula siempre se ajusta al ancho y se desplaza en vertical, media fila por muesca de la rueda en todos los hosts. WinForms se desplaza mediante `ScrollableControl`. WPF coloca la cuadrícula en un `ScrollViewer`. Avalonia muestra su propia `ScrollBar` y expone la posición como **VerticalOffset**, que mueve los tiles hacia arriba y los recorta al dashboard.

Eventos

OnLayoutChanged es una propiedad del tipo delegado con nombre de Delphi **TsgcUINotifyEvent**, como en los demás productos sgc .NET: asígnela con `=`, contiene un solo manejador. Los orígenes del puente se suscriben a los eventos de sgcWebSockets .NET con `+=` y solo quitan su propio manejador, de modo que los manejadores de la aplicación siguen disparándose, véase Feeds en vivo en .NET.

Sin verbos de diseñador y sin tile de mapa

El verbo del editor de componentes de Delphi "Add Tile" no tiene equivalente en WinForms: añada los tiles en código, o declárelos en XAML en WPF y Avalonia. El ajuste en tiempo de diseño de un tile movido tampoco forma parte del port. La demo Suite de Delphi tiene un tile de sgcMaps; sgcMaps no tiene port .NET, por lo que la demo Suite de .NET tiene un gráfico, un visor de PDF y una vista previa de informe, pero ningún mapa.

Ediciones

Los paquetes Community se compilan con `SGC_COMMUNITY`. El arrastre y el redimensionado en tiempo de ejecución, **Locked**, la persistencia de la disposición y **OnLayoutChanged** se excluyen de la compilación, y el paquete del puente no se genera, como el bloque `SGC_DASHBOARD_PERSIST` de Delphi. Proteja el código que los usa con `#if !SGC_COMMUNITY`.

Enlace de datos

En WPF y Avalonia las propiedades visibles para el usuario son propiedades de dependencia y propiedades de Avalonia, por lo que se pueden enlazar, aplicar con estilos y animar, consulte Enlace de datos en .NET.

Véase también

- Visión general de .NET
- Clases de .NET
- Renderizado sin interfaz en .NET
- Feeds en vivo en .NET
- Disposición
- Temas
- Persistencia

Enlace de datos en .NET

En los hosts WPF y Avalonia, las propiedades visibles para el usuario del dashboard, de sus tiles y de la tarjeta KPI son propiedades de dependencia (WPF) y propiedades de Avalonia, por lo que se pueden enlazar a un modelo de vista, establecer mediante un Style y animar. Este tema se aplica solo a WPF y Avalonia. Los controles WinForms no cambian: sus propiedades son propiedades CLR simples, como antes.

Propiedades enlazables

Cada propiedad de la tabla tiene un campo estático con el sufijo Property, por ejemplo `TsgcWpfDashboardTile.ColProperty`. En WPF los campos son DependencyProperty. En Avalonia son StyledProperty, excepto Col, Row, ColSpan y RowSpan, que son DirectProperty. Las propiedades marcadas con **TwoWay** se enlazan TwoWay de forma predeterminada, porque un arrastre o un cambio de tamaño en tiempo de ejecución y LoadLayout las modifican, de modo que un enlace sin Mode mantiene actualizado el modelo de vista. Las demás propiedades se enlazan OneWay de forma predeterminada.

Clase WPF	Clase Avalonia	Propiedades enlazables
TsgcWpfDashboard	TsgcAvaloniaDashboard	Columns, RowHeight, Gap, Padding, AutoFlow, Theme, AllowDrag, AllowResize
TsgcWpfDashboardTile	TsgcAvaloniaDashboardTile	Title, **Col (TwoWay)**, **Row (TwoWay)**, **ColSpan (TwoWay)**, **RowSpan (TwoWay)**, ShowHeader, ContentPadding, ParentTheme, Locked
TsgcWpfKPICard	TsgcAvaloniaKPICard	Title, Value, Icon, Trend, TrendValue, FooterText, AccentColor, MaxPoints

AllowDrag y AllowResize no están en la edición Community. Padding es el espacio alrededor de la cuadrícula en unidades independientes del dispositivo, un int. En Avalonia, Theme es el tema de sgcUICore (TsgcUITheme) y oculta el tema de control de Avalonia de StyledElement, que sigue accesible mediante `StyledElement.ThemeProperty`.

Comportamiento

- **Un arrastre escribe la celda de vuelta.** Un arrastre o un cambio de tamaño en tiempo de ejecución, y LoadLayout, cambian la celda de un tile, y el tile copia la celda ajustada en Col, Row, ColSpan y RowSpan. Con el enlace TwoWay predeterminado, el modelo de vista recibe la celda donde quedó el tile.
- **Una celda enlazada es un cambio de diseño.** Una nueva celda que llega a través de un enlace de datos se trata como un arrastre, por lo que se genera OnLayoutChanged del dashboard (no en la edición Community). Establecidas desde el código (los setters CLR o SetCell), las propiedades de celda se comportan como antes.
- **Un valor rechazado se escribe de vuelta.** Un valor que el dashboard o el tile no acepta (Columns o RowHeight inferior a 1, un Gap o Padding negativo, un span inferior a 1, un ContentPadding negativo) no se aplica, y el valor en uso se escribe de vuelta en la propiedad y en un origen TwoWay, en WPF y en Avalonia. Lo mismo se aplica a un valor que la tarjeta KPI corrige (MaxPoints inferior a 1, un texto null). En WPF el origen recibe el valor en uso cuando el dispatcher ha ejecutado el trabajo pendiente. En Avalonia la escritura de vuelta cubre un valor que el origen envía cuando el enlace ya está en marcha. Un valor que el origen ya tiene cuando el enlace se inicia se corrige en el control, pero no se escribe de vuelta en el origen.
- **LoadLayout actualiza las propiedades.** Tras LoadLayout, las métricas de la cuadrícula y las celdas de los tiles se copian de vuelta en las propiedades, de modo que los enlaces ven el diseño restaurado.
- **Styles.** Un setter de Style puede establecer cualquier propiedad enlazable. Un Theme del dashboard establecido mediante un Style llega a todos los tiles cuyo ParentTheme es true.

Propiedades que siguen siendo propiedades CLR

Propiedad	Motivo
Tiles (dashboard)	La colección de tiles, rellenada por los tiles declarados en XAML, no un valor que enlazar.
Layout (dashboard)	El objeto del motor de diseño, de solo lectura. Sus métricas de cuadrícula se enlazan mediante Columns, RowHeight, Gap, Padding y AutoFlow.
VerticalOffset (dashboard de Avalonia)	La posición de desplazamiento de la última pasada de diseño, no un ajuste.
Theme (tile)	Mientras ParentTheme es true, el tile lee el tema del dashboard, y establecer Theme borra ParentTheme. Enlace en su lugar el Theme del dashboard, o ParentTheme.
Theme (tarjeta KPI)	Lo establece el tile que aloja la tarjeta.

Ejemplo

El modelo de vista siguiente se comparte en ambos ejemplos. Revenue está enlazado al Value de una tarjeta KPI, y RevenueCol y RevenueRow están enlazados a Col y Row de su tile sin Mode, por lo que se enlazan TwoWay: cuando el usuario arrastra el tile, el modelo de vista recibe la nueva celda. Establezca AllowDrag en True en el dashboard para que el usuario pueda arrastrar los tiles.

WPF

MainWindow.xaml crea el modelo de vista como DataContext de la ventana. El código subyacente y App.xaml son los de la plantilla WPF.

Avalonia

MainWindow.axaml establece `x:DataType` con el modelo de vista, por lo que los enlaces son enlaces compilados. El código subyacente llama a InitializeComponent, y la clase App es la de Inicio rápido Avalonia.

Consulte también

- Diferencias de .NET respecto a Delphi
- Clases de .NET
- Inicio rápido WPF
- Inicio rápido Avalonia

Renderizado sin interfaz en .NET

El motor de disposición y la tarjeta KPI no necesitan ningún framework de interfaz, por lo que un servidor, una aplicación de consola o una prueba pueden colocar tiles y renderizar tarjetas en archivos de imagen. Es el equivalente en .NET de Imágenes KPI sin interfaz.

Métodos

Método	Paquete	Descripción
TsgcDashboardKPI.SaveToSVG(stream or file, width, height, scale)	esegece.sgcDashboard	Escribe la tarjeta como texto SVG, byte a byte el SVG de la tarjeta de Delphi.
TsgcDashboardKPI.SaveToPNG(stream or file, width, height, scale)	esegece.sgcDashboard.Rendering	Un método de extensión que escribe un PNG con fondo transparente. La imagen mide ancho por escala por alto por escala píxeles.
TsgcDashboardKPI.Render(canvas, width, height, scale)	esegece.sgcDashboard	Dibuja la tarjeta en cualquier lienzo de sgcUI-Core.
TsgcDashboardLayout.Arrange(width, scale)	esegece.sgcDashboard	Coloca los elementos; **PlacedCell**, **Rect** y **ContentHeight** devuelven el resultado.
TsgcDashboardLayout.SaveToJSON, LoadFromJSON	esegece.sgcDashboard	El texto de disposición de los dashboards. Edición Registered.

Ejemplo

El programa escribe revenue.svg y revenue.png, de 560 por 400 píxeles, e imprime:

```
revenue: column 0, row 0
orders: column 2, row 2
{"version":1,"columns":12,"rowHeight":100,"gap":12,"padding":12,"tiles":[{"name":"revenue","col":0,"row":0,"colSpan":4,"row
orders back at column 2, row 0
```

El tile orders pide la columna 2, fila 0, que se solapa con el tile revenue, por lo que el motor lo empuja hacia abajo a la fila 2. **Cell** devuelve la celda que pidió un tile, **PlacedCell** la celda que obtuvo; el JSON guarda las celdas solicitadas, como Delphi.
Los paquetes sin interfaz funcionan en Windows, Linux y macOS. En Linux, esegece.sgcDashboard.Rendering incorpora los recursos nativos de SkiaSharp para Linux.

Véase también

- Visión general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Imágenes KPI sin interfaz

- TsgcDashboardKPI
- Disposición

Feeds en vivo en .NET

El paquete esegece.sgcDashboard.sgcWebSockets alimenta tarjetas KPI desde una conexión en vivo de sgcWebSockets .NET, como el puente sgcDashboardWS de Delphi, véase Datos en vivo. Un origen lee un número de cada mensaje JSON y lo envía a una tarjeta: el sparkline recibe el punto, y se actualizan **Value**, **Title**, **TrendValue** y **Trend**. **Solo en la edición Registered**: el paquete del puente no se distribuye en la edición Community.

Orígenes

Origen	Cliente	Lee
TsgcDashboardSource_WebSocket	TsgcWebSocketClient	Cada mensaje de texto, mediante **OnMessage**.
TsgcDashboardSource_MQTT	TsgcWSProtocol_MQTT_Client	Cada publish cuyo topic coincide con **Topic**, con los comodines de MQTT `+` y `#`, mediante **OnMQTTPublish**. El origen no se suscribe; la aplicación se suscribe a través del cliente.
TsgcDashboardSource_SSE	TsgcHTTP1Client en **Client**, o TsgcSSEClient en **SSEClient**	Los datos de cada server sent event, a partir de los fragmentos de **GetSSE** y **OnSSEMessage**, o a partir de los eventos de **OnEvent**. **GetSSE** bloquea hasta que termina el flujo, por lo que debe llamarse desde un hilo de trabajo.

Todos los orígenes comparten los miembros de TsgcDashboardSource:

Miembro	Descripción
Card	Un control de tarjeta KPI (cualquier **IsgcDashboardKPIHost**) o un **TsgcDashboardKPI** sin interfaz. Cualquier otro objeto genera **EsgcDashboardSource**.
ValuePath	Ruta JSON del número, por ejemplo `price` o `data.items[0].value`. Se lee un número o una cadena numérica, con punto como separador decimal. Vacío lee el mensaje completo como el número.
TitlePath, TrendValuePath	Cuando se establecen, copian ese valor JSON en **Title** y **TrendValue** de la tarjeta.
ValueFormat	Una cadena de formato de Delphi con un argumento float, como `%.2f` o `%.1f ms`, formateada con la cultura actual. Vacío escribe el número como el FloatToStr de Delphi.
Active	True por defecto. False ignora todos los mensajes.
OnValue	Se llama con cada número leído, después de actualizar la tarjeta.

Subprocesos

El punto se añade al sparkline de inmediato, en el hilo en el que llega el mensaje. Los textos y la tendencia (al alza, a la baja o neutra respecto al valor anterior del origen) se aplican mediante el `SynchronizationContext` que esta-

ba activo cuando se creó el origen, el hilo de interfaz de una aplicación WinForms, WPF o Avalonia, por lo que cualquier modo **NotifyEvents** del cliente es seguro. Un origen creado sin contexto de sincronización, en una aplicación de consola o un servicio, aplica la actualización en el hilo del mensaje.

Ejemplo

Tres tarjetas KPI en un dashboard WinForms, alimentadas por un flujo de cotizaciones WebSocket, un topic de sensor MQTT y un feed de server sent events. Los nombres de host son marcadores de posición para sus propios servidores.

Pruébelo en la interfaz de loopback

Este programa de consola inicia un **TsgcWebSocketServer** en el mismo proceso en la interfaz de loopback, conecta a él un **TsgcWebSocketClient** y alimenta una tarjeta sin interfaz con tres mensajes de broadcast, por lo que no necesita red.
Imprime el valor formateado en la cultura actual, aquí con coma como separador decimal, la tendencia respecto al valor anterior y el número de puntos del sparkline:

```
EUR/USD 1,0839, trend ktDown, 3 points
```

Diferencias respecto a Delphi

- **OnMessage**, **OnMQTTPublish**, **OnSSEMessage** y **OnEvent** son eventos de .NET. Un origen añade su manejador con `+=` y quita solo su propio manejador, mientras que Delphi guarda, reemplaza y restaura el manejador de la aplicación. Los manejadores de la aplicación siempre se disparan, en el orden de suscripción.
- En .NET no existe FreeNotification. Cuando la tarjeta es un `IComponent`, como un control WinForms, el origen borra **Card** cuando se elimina la tarjeta. Para cualquier otra tarjeta establezca **Card** a null antes de que la tarjeta desaparezca.
- **SSEClient** del origen SSE, que acepta un **TsgcSSEClient**, es una adición de .NET.

Véase también

- Visión general de .NET
- Instalación desde NuGet
- Diferencias de .NET respecto a Delphi
- Datos en vivo
- TsgcDashboardSource
- TsgcDashboardSource_WebSocket
- TsgcDashboardSource_MQTT
- TsgcDashboardSource_SSE
- Demos de .NET

Demos de .NET

Las demos de .NET están en la carpeta `demos` del producto y en el zip de demos, con la solución `sgcDashboard_Demos.sln`. Cada demo tiene un README.md.

Demo	Host	Descripción
Basic	WinForms	Seis tiles en una cuadrícula de 12 columnas: cuatro **TsgcWinFormsKPICard** alimentadas por un temporizador con un paseo aleatorio, un registro de eventos y una lista de servidores en controles comunes, y un combo Theme que cambia el dashboard entre claro y oscuro.
BasicWpf	WPF	El mismo dashboard declarado en XAML sobre un **TsgcWpfDashboard**.
BasicAvalonia	Avalonia	El mismo dashboard sobre un **TsgcAvaloniaDashboard**, en Windows, Linux y macOS.
Suite	WinForms	Los productos de eSeGeCe en un solo dashboard: dos tarjetas KPI, un gráfico de sgcCharts con una línea en vivo, un visor de sgcPDF y una vista previa de sgcReports en tiles, y un conmutador de claro y oscuro. No hay tile de mapa, porque sgcMaps no tiene port .NET.
LiveKPI	WinForms	Tres tarjetas KPI alimentadas por **TsgcDashboardSource_WebSocket** desde un **TsgcWebSocketServer** en el mismo proceso en un puerto de loopback libre. Solo en la edición Registered.
SaveLayout	WinForms	Arrastre y redimensione los tiles, luego guarde la disposición en layout.json, cárguela de nuevo o restablézcala. Solo en la edición Registered.

Con los paquetes Community, LiveKPI y SaveLayout siguen compilando: excluyen el código Registered de la compilación y explican el límite en la barra de estado.

Compilar y ejecutar

Las demos necesitan el SDK de .NET 8 o posterior. Ejecute una demo desde su carpeta:

```
dotnet run
```

Dentro del árbol del producto las demos se compilan contra los proyectos de la biblioteca. Desde el zip de demos restauran los paquetes NuGet, la edición Community por defecto. Establezca **SgcDashboardEdition** para usar los paquetes Registered:

```
dotnet build -p:SgcDashboardEdition=Registered
```

Autocomprobación

Toda demo tiene un conmutador `--test` que abre la ventana fuera de pantalla y la controla: Basic comprueba los tiles y sus celdas, alimenta las tarjetas y cambia el tema, Suite comprueba el gráfico, el visor de PDF y la vista previa de informe en sus tiles, LiveKPI comprueba que cada tarjeta ha recibido valores, y SaveLayout guarda, mueve un tile, carga y restablece. La demo escribe una línea PASS o FAIL por comprobación en el archivo y termina con el número de comprobaciones fallidas, por lo que 0 significa que todas pasaron.

```
dotnet run -- --test result.txt
```

Véase también

- Visión general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Feeds en vivo en .NET
- Demos de Delphi

Solución de problemas

Identificador no declarado SaveLayout, AllowDrag u OnLayoutChanged

La edición Community se compila sin SGC_DASHBOARD_PERSIST, por lo que AllowDrag, AllowResize, OnLayoutChanged, los métodos SaveLayout y LoadLayout y la propiedad Locked de VCL no existen, y el código que los usa falla con E2003 Undeclared identifier. Un formulario guardado por una build Registered que tiene AllowDrag o AllowResize en su .dfm lanza una excepción "Error reading" con "Property AllowDrag does not exist". Elimine esos miembros o use una build Registered o Trial. Consulte Ediciones.

No se encuentra la unidad sgcDashboardWS_Source_WebSocket o sgcWebSocket

El puente de datos en vivo es un paquete independiente, sgcDashboardWS, que requiere sgcDashboard y sgcWebSockets. Instale primero sgcWebSockets para el mismo IDE, después los paquetes sgcDashboardWS, y añada la carpeta de fuentes o de bibliotecas de sgcWebSockets a la ruta de búsqueda. El puente no se distribuye en la edición Community.

Una tarjeta KPI alimentada por un origen del puente nunca cambia en una aplicación de consola o servicio

En Delphi 2010 y posteriores, los textos de la tarjeta se aplican en el hilo principal mediante TThread.Queue. Una aplicación de consola o un servicio no tiene un bucle de mensajes que ejecute la cola, por lo que debe llamar a CheckSynchronize con regularidad. Los valores del sparkline se añaden de inmediato en cualquier caso.

El Value de una tarjeta muestra una coma como separador decimal

ValueFormat se aplica con Format, que usa el separador decimal de la configuración regional de la aplicación. El número JSON en sí se lee siempre con punto. Cambie ValueFormat, o establezca el valor desde OnValue con su propio formato.

El dashboard FMX no compila para Linux64

Los controles FMX en Linux necesitan FMX para Linux (FMXLinux) instalado en el IDE. Sin él, en Linux64 solo se compilan las unidades independientes del framework: el motor de disposición y el modelo KPI con su salida PNG y SVG sin interfaz. Consulte Plataformas.

Un control alojado conserva sus colores cuando cambia el tema

El tema llega a un control alojado solo mediante un Theme, Chart.Theme, BackgroundColor o Color publicado. La vista previa de sgcReports no publica ninguno de ellos y conserva sus propios colores, y los controles anidados dentro de un panel de un tile no reciben el tema. Theme también se ignora con utCustom. Consulte Temas.

La demo Suite o LiveKPI no compila

La demo Suite necesita los fuentes de sgcCharts, sgcMaps, sgcPDF y sgcReports en la ruta de búsqueda, y LiveKPI necesita los fuentes de sgcWebSockets. Consulte Demos.

Introducción

sgcGrid es una cuadrícula de datos en vivo para Delphi y C++Builder, para VCL y FireMonkey. Está pensada para datos que cambian constantemente: las filas se actualizan por clave desde cualquier hilo, las celdas modificadas destellan en verde o en rojo y se desvanecen, las ráfagas de actualizaciones se agrupan en repintados temporizados, y el modo cola mantiene a la vista las filas más recientes de un registro o de un flujo. **TsgcVCLGrid** y **TsgcFMXGrid** ordenan por varias columnas, filtran, seleccionan filas o celdas, cambian el tamaño de las columnas, aplican reglas de formato condicional y editan celdas en el lugar.
sgcGrid no es una cuadrícula de base de datos de propósito general con bandas, agrupación y tablas dinámicas. Es una cuadrícula especializada que se vende por su rendimiento: solo dibuja las filas visibles y un fotograma completo de 1080p de una cuadrícula de 100.000 filas tarda unos pocos milisegundos. Consulte Rendimiento.
sgcGrid forma parte de la suite visual de eSeGeCe. Se dibuja con **sgcUICore**, el runtime compartido con el lienzo, el rasterizador y los temas claro y oscuro. Sus datos implementan la interfaz de tabla de sgcUICore IsgcUITable, de modo que un gráfico de **sgcCharts** puede representar las filas de una cuadrícula con un TsgcChartDataSource_Table, y una cuadrícula colocada en un mosaico de **sgcDashboard** sigue el tema del panel.

Symbol	Price ▼	Change %	Volume	Halted
MSFT	412.50	+1.25	1,200,500	
AAPL	228.10	-0.84	3,400,200	
AMZN	186.40	-2.10	2,250,000	✓
GOOG	171.90	+0.37	980,100	
NVDA	121.70	+3.02	**5,120,750**	

Ediciones

sgcGrid se distribuye como Registered, Trial y Community. La edición Community excluye la edición en el lugar (SGC_GRID_EDIT) y el puente sgcGridWS con los flujos en vivo y la consulta de IA. Consulte Ediciones.

Por dónde empezar

- Arquitectura: el modelo independiente del framework, el renderizador sobre el lienzo de sgcUICore, los controles VCL y FMX y el modelo de subprocesos.
- Plataformas y Rendimiento: las versiones de Delphi, C++Builder, las plataformas y el presupuesto de fotograma.
- Inicio rápido VCL, Inicio rápido FMX e Inicio rápido C++Builder: una primera cuadrícula en pocos minutos.
- Componentes: las cuadrículas, las clases del modelo, los orígenes de datos y la consulta de IA.
- Actualizaciones en vivo y subprocesos, Ordenación, Filtrado, Selección y teclado y Cambio de tamaño de columnas.
- Destello al cambiar, Formato condicional, Edición en el lugar y Modo virtual.
- JSON y CSV, Enlace con DataSet, Flujos en vivo, Pregunte a su cuadrícula y Gráficos desde la cuadrícula.
- Temas y Renderizado sin interfaz.
- Demos: los proyectos de demostración de Delphi y C++Builder.

Arquitectura

sgcGrid está construido en capas. Cada capa solo depende de las inferiores, de modo que el modelo, el renderizador y las unidades de importación y exportación se compilan sin VCL ni FireMonkey y también se prueban en Linux64.

Capa	Unidad	Función
Modelo	sgcGrid_Columns, sgcGrid_Data	TsgcGridData contiene las columnas y las filas: un Variant por columna, una marca de cambio y una dirección de cambio por celda, indexados por clave de fila. Ordena, filtra y admite un modo virtual. Es seguro para subprocesos e implementa IsgcUITable.
Vista de líneas	sgcGrid_Lines, sgcGrid_Group, sgcGrid_Summary	TsgcGridLineView asigna las líneas de visualización de una cuadrícula (filas de grupo y filas de la vista) a las filas de la vista de los datos, TsgcGridGrouper construye los grupos y sgcGrid_Summary calcula los resúmenes de grupo y de pie. En la edición Registered, sgcGrid_Tree y sgcGrid_Detail añaden las filas de árbol y las bandas de detalle.
Formato	sgcGrid_Format	El destello al cambiar y las reglas de formato condicional, compartidos por todos los renderizadores.
Renderer	sgcGrid_Render	TsgcGridRenderer dibuja un TsgcGridData en cualquier lienzo de sgcUICore: encabezado, filas, líneas de cuadrícula, selección, glifos de ordenación, marcas de verificación y los estilos de celda. Solo se dibujan las filas visibles. También escribe archivos PNG y SVG sin ventana.
Importación y exportación	sgcGrid_IO, sgcGrid_Source_DataSet	Carga y guardado de JSON y CSV, y el TsgcGridSource_DataSet que rellena una cuadrícula desde cualquier TDataSet.
Exportación a Excel	sgcGrid_Zip, sgcGrid_XLSX	El escritor de XLSX que hay detrás de SaveToXLSX, solo en la edición Registered. Vea Exportación a Excel.
Control VCL	sgcGrid_VCL	TsgcVCLGrid posee un TsgcGridData y un renderizador, y añade las barras de desplazamiento, el ratón, el teclado, el portapapeles y los editores en el lugar.
Control FMX	sgcGrid_FMX	TsgcFMXGrid es el control de FireMonkey con las mismas propiedades, métodos y eventos, barras de desplazamiento dibujadas y desplazamiento táctil.
Puente	sgcGridWS_*	Los orígenes de flujo WebSocket, MQTT y SSE y TsgcGridAIQuery. Necesitan sgcWebSockets.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme, sgcUICore_Table	El runtime compartido: el lienzo vectorial y el rasterizador, la base de los controles VCL y FMX con control de fotogramas (MaxFPS), los tokens de tema y la interfaz IsgcUITable.

Modelo de subprocesos

Todo lector y escritor público de TsgcGridData toma una sección crítica reentrante interna, de modo que un hilo de flujo puede llamar directamente a UpdateRow, AppendRow, UpdateCell y DeleteRow, sin Synchronize. Los datos lanzan su OnChange en el hilo escritor una vez liberado el bloqueo. La cuadrícula no dibuja allí: marca su contenido como sucio y la base de control de sgcUICore programa un repintado en el hilo principal, como máximo **MaxFPS** veces por segundo (60 de forma predeterminada). Los cambios de vista, como una fila nueva o una reordenación, se envían una vez al hilo principal, donde se actualizan el rango de desplazamiento, la posición de cola y la fila seleccionada, localizada de nuevo por su clave.
BeginUpdate y EndUpdate agrupan cualquier número de cambios en un único OnChange. Las columnas se modifican desde el hilo principal; mientras se ejecutan hilos escritores, envuelva los cambios de columnas en Lock y Unlock. Consulte Actualizaciones en vivo y subprocesos.

Vista y almacén

Las filas se almacenan en orden de inserción. Sin columnas de ordenación y sin filtro, la cuadrícula muestra el almacén tal cual. En caso contrario, la vista se reconstruye de forma diferida en la primera lectura tras un cambio: primero el filtro y después una ordenación estable por varias columnas. Un cambio en una columna que no está ordenada, sin filtro activo, conserva la vista actual, de modo que una columna de precios que cambia constantemente no cuesta una reordenación.

Vista de líneas

Las cuadrículas pintan y desplazan líneas de visualización, no filas de la vista. Cada cuadrícula tiene un TsgcGridLineView sobre sus datos. Sin ninguna columna agrupada, la vista de líneas es la identidad, la línea i es la fila de la vista i, y no se construye nada. La agrupación y, en la edición Registered, las filas de árbol y las bandas de detalle, son estado de la vista de líneas: los datos, sus filas de la vista e IsgcUITable permanecen planos, de modo que los gráficos, las cuadrículas dinámicas y el puente ven las mismas filas que antes. La vista de líneas se reconstruye de forma diferida cuando el LayoutVersion de los datos se mueve, y solo actualiza los resúmenes de grupo cuando únicamente han cambiado valores. Vea Agrupación.

Consulte también

- Plataformas
- Rendimiento
- TsgcGridData
- TsgcGridRenderer
- TsgcGridLineView

Ediciones

sgcGrid se compila desde el mismo código fuente para todas las ediciones. La edición solo cambia qué características compila una compilación y qué unidades se distribuyen.

Edición	Incluye
Registered	Todo: las cuadrículas VCL y FMX, el modelo, la ordenación, el filtrado, la agrupación con el panel de grupos, los resúmenes de grupo y de pie, las columnas fijas, las filas de árbol, las filas maestro detalle, la exportación a Excel, la selección, el parpadeo, las reglas de formato, la edición en el sitio, el modo virtual, JSON y CSV, el enlace con DataSet, los temas, la salida PNG y SVG sin interfaz, y el puente sgcGridWS con las fuentes de datos WebSocket, MQTT y SSE y la consulta con IA.
Trial	Las mismas características que Registered. Una compilación con SGC_TRIAL muestra un aviso de que es una versión de prueba cuando ha pasado la ventana de versión indicada en la compilación. No cambia nada más.
Community	Todo salvo la edición en el sitio, las filas de árbol, las filas maestro detalle, la exportación a Excel y el puente sgcGridWS.

Lo que Community deja fuera

- La edición en el lugar. La edición Community se compila con SGC_GRID_EDIT sin definir, por lo que StartEdit, CommitEdit, CancelEdit, EditorActive y los eventos OnBeforeEdit, OnValidate, OnEdited y OnEditError no existen, y ReadOnly False no tiene editor. Consulte Edición en el lugar.
- Filas de árbol, filas maestro detalle y exportación a Excel. La edición Community se compila con SGC_GRID_ADVANCED sin definir, de modo que las propiedades Tree y Detail, ExpandNode, CollapseNode, ExpandDetail, CollapseDetail, DetailExpanded, DetailData, SaveToXLSX, los eventos OnTreeExpandChange, OnDetailExpandChange, OnGetDetailData y OnDetailCellClick y el componente TsgcGridDataStore no existen. Vea Filas de árbol, Maestro detalle y Exportación a Excel.
- El puente sgcGridWS: las unidades sgcGridWS_*.pas y sus paquetes no se distribuyen, por lo que los orígenes de flujo y TsgcGridAIQuery no están disponibles. Consulte Flujos en vivo y Pregunte a su cuadrícula.

La agrupación, el panel de grupos, los resúmenes de grupo y de pie y las columnas fijas forman parte de todas las ediciones, incluida Community. Vea Agrupación, Resúmenes y Columnas fijas.
La edición es una decisión en tiempo de compilación en sgcVer.inc (SGC_COMMUNITY, que también desactiva SGC_TRIAL), no un interruptor en tiempo de ejecución. El código que usa la API de edición, los miembros de árbol, detalle y Excel o el puente no compila con una compilación Community, vea Solución de problemas.

Instalación

sgcGrid forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcGrid** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcGrid

- **sgcUICore**, la base compartida de los productos de disposición y de sgcCharts. No tiene casilla propia. El instalador la instala automáticamente siempre que se selecciona sgcGrid.
- **sgcGrid para sgcWebSockets**, que aparece en **sgcWebSockets bridges**. Instala los paquetes sgcGridWS con los orígenes de flujo y TsgcGridAIQuery (consulte Flujos en vivo). El instalador solo lo ofrece cuando sgcWebSockets ya está instalado en el equipo y su licencia incluye sgcGrid. De lo contrario aparece atenuado.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Grid`	Las demos de sgcGrid. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluido sgcGrid.
- **Community**: el instalador gratuito Community de sgcUI incluye sgcGrid, junto con sgcCharts, sgcDashboard, sgcNavigation, sgcToolbar y sgcWizard. Los puentes de sgcWebSockets no forman parte del instalador Community, y sgcPDF y sgcMaps no forman parte de Community. Consulte Ediciones para saber qué deja fuera la compilación Community.

Instalaciones independientes anteriores

Si el instalador encuentra una instalación independiente anterior de sgcUICore, como la que escribió el instalador independiente de sgcCharts, muestra las rutas de bibliotecas y los paquetes de diseño de esa instalación en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcGridD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcGrid<V>` / `dclsgcGrid<V>`	Los componentes de sgcGrid (tiempo de ejecución / tiempo de diseño).
`sgcGridWS<V>` / `dclsgcGridWS<V>`	El puente de sgcWebSockets (tiempo de ejecución / tiempo de diseño). Requiere sgcWebSockets.

Compílelos en orden de dependencia: `sgcUICore`, después `sgcGrid`, después `dclsgcGrid` y, por último, los paquetes del puente. Después instale los paquetes de diseño `dcl*` y añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` correspondiente a su versión, por ejemplo `sgcGridC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

.NET

No cambia nada para .NET. sgcGrid se sigue distribuyendo como sus propios paquetes NuGet (`esegece.sgcGrid` y sus paquetes complementarios). Consulte la instalación de .NET.

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13, con paquetes de runtime independientes del framework, VCL y FireMonkey y paquetes de tiempo de diseño VCL y FireMonkey, véase Paquetes más abajo.
C++Builder	De C++Builder 2007 a C++Builder 13, con los encabezados .hpp generados y las bibliotecas estáticas. Consulte Inicio rápido C++Builder.
VCL	TsgcVCLGrid en Windows, Win32 y, en las versiones que lo admiten, Win64.
FireMonkey	TsgcFMXGrid desde Delphi XE6 (SGC_GRID_FMX), en Windows, macOS, iOS y Android.
Linux64	Las unidades independientes del framework (el modelo, el formato, el renderizador y la importación y exportación de JSON y CSV) desde Delphi 10.2 (SGC_GRID_LINUX), por ejemplo para renderizar imágenes de cuadrículas en un servidor.
Puente	Los paquetes sgcGridWS requieren el paquete sgcGrid y el paquete sgcWebSockets de la misma versión de Delphi o C++Builder.
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, la salida sin interfaz en PNG y SVG, el origen DataTable y los puentes con sgcWebSockets y sgcHTML, sobre .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Paquetes

sgcGrid incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcGridD13.dpk para Delphi 13 y sgcGridC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcGrid	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey: el modelo, las columnas, el formato, el renderizador, la agrupación, los resúmenes, las filas de árbol y de detalle, la importación y exportación de JSON, CSV y XLSX, TsgcGridDataStore y TsgcGridSource_DataSet.	rtl, dbrtl, sgcUICore
sgcGridVCL	Runtime, VCL	sgcGrid_VCL (TsgcVCLGrid)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcGrid
sgcGridFMX	Runtime, FireMonkey, desde XE6	sgcGrid_FMX (TsgcFMXGrid)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcGrid

dclsgcGrid	Tiempo de diseño, VCL	Registra TsgcGridSource_DataSet, TsgcGridDataStore (no en la edición Community) y TsgcVCLGrid en la página de paleta **SGC Grid**, y un editor de propiedades para la Source de las opciones de detalle, que solo lista los componentes que contienen datos de la rejilla.	designide, rtl, vcl, sgcGrid, sgcGridVCL
dclsgcGridFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXGrid en la página de paleta **SGC Grid**.	designide, rtl, fmx, sgcGrid, sgcGridFMX, dclsgcGrid
sgcGridWS, dclsgcGridWS	Runtime y tiempo de diseño, puente de sgcWebSockets	Las fuentes de datos en vivo (TsgcGridFeedSource_WebSocket, TsgcGridFeedSource_MQTT, TsgcGridFeedSource_SSE) y TsgcGridAIQuery, independientes del framework, en la página de paleta **SGC Grid**.	rtl, sgcGrid, sgcWebSockets

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcGridFMX requiere dclsgcGrid, así que instale primero dclsgcGrid. El instalador instala sgcUICore junto con sgcGrid. Si compila los paquetes usted mismo, compile primero sgcUICore.

Versiones del IDE y plataformas de destino

Delphi	sgcGrid	sgcGridVCL	sgcGridFMX	dclsgcGrid, dclsgcGridFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE. sgcGridWS se compila para las mismas plataformas que sgcGrid, y dclsgcGridWS para las mismas plataformas que dclsgcGrid.

C++Builder	sgcGrid, sgcGridVCL	sgcGridFMX	dclsgcGrid, dclsgcGridFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcGrid)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcGrid)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcGridC13.cbproj, sgcGridVCLC13.cbproj, sgcGridFMXC13.cbproj, dclsgcGridC13.cbproj y dclsgcGridFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. sgcGridWS y dclsgcGridWS se compilan para las mismas versiones de C++Builder y plataformas que sgcGrid y dclsgcGrid, excepto Win64x, porque necesitan una compilación Win64x del paquete sgcWebSockets. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXGrid) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcGridVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcGrid_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcGrid se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa el modelo, el formato, el renderizador y la importación y exportación de JSON y CSV, por ejemplo para renderizar imágenes de la rejilla. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Consulte también

- Ediciones
- Arquitectura

Rendimiento

La cuadrícula solo dibuja las filas y las columnas visibles, toma el bloqueo de los datos una vez por fotograma, almacena en caché la geometría de las columnas y dibuja el texto de las celdas desde una caché de imágenes de glifos en el lienzo de trama. El coste de un fotograma depende del tamaño de la ventana, no del número de filas.

Presupuesto de fotograma

La suite de QA tiene una prueba de rendimiento: un fotograma de 1920 x 1080 con 12 columnas, 100.000 filas y 45 filas visibles, renderizado 60 veces en el lienzo de trama. La prueba falla por encima de 16 ms por fotograma en Win64 y 25 ms en Win32. Los últimos promedios medidos fueron:

Plataforma	Fotograma simple	Destello y 3 reglas de formato
Win32	3.0 ms	3.2 ms
Win64	4.6 ms	6.8 ms
Linux64	3.7 ms	

Así, una cuadrícula de 1080p se renderiza en unos pocos milisegundos, muy por debajo de los 16,7 ms de un fotograma a 60 fps.

Frecuencia de actualización

Las actualizaciones y los fotogramas están desacoplados. Los escritores pueden llamar a UpdateRow con la frecuencia con que el flujo entregue datos, y la cuadrícula se repinta como máximo MaxFPS veces por segundo, sin importar cuántas filas hayan cambiado entretanto. La demo LiveTicker escribe unas 1.000 actualizaciones de precio por segundo desde un hilo de trabajo en una cuadrícula ordenada y con destellos a 60 fps.

Consejos

- Envuelva un lote de cambios en BeginUpdate y EndUpdate para que la cuadrícula reciba una única notificación.
- Ordenar por una columna que cambia constantemente reordena la vista con cada cambio de esa columna. Los cambios en columnas que no están ordenadas, sin filtro, conservan la vista actual.
- Use MaxRows para registros y flujos que crecen sin fin, y el modo virtual para datos muy grandes que ya residen en otro lugar.
- Reduzca MaxFPS cuando la cuadrícula solo necesite parecer en vivo, no fluida.

Consulte también

- Actualizaciones en vivo y subprocesos
- Modo virtual
- Arquitectura

Inicio rápido VCL

En tiempo de diseño

- Coloque un **TsgcVCLGrid** de la página de paleta sgcGrid en un formulario y establezca Align en alClient.
- Abra el editor de **Columns** y añada una columna por campo. Establezca FieldName, Caption y DataType, y DisplayFormat para números y fechas.
- Rellene las filas desde código con Data.AppendRow y manténgalas al día con Data.UpdateRow, desde cualquier hilo.

Desde código

```
uses
  Forms, Classes, Controls,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_VCL;

procedure CreateGrid(aForm: TForm);
var
  oGrid: TsgcVCLGrid;
begin
  oGrid := TsgcVCLGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := alClient;

  // the columns: FieldName, Caption and DataType
  oGrid.Columns.Add('Symbol', 'Symbol');
  oGrid.Columns.Add('Price', 'Price', gdtFloat).DisplayFormat := '0.00';
  oGrid.Columns.Add('Volume', 'Volume', gdtInteger);

  // one row per key, the values in column order
  oGrid.Data.AppendRow('MSFT', ['MSFT', 412.5, 1200]);
  oGrid.Data.AppendRow('AAPL', ['AAPL', 228.1, 3400]);

  // later, from any thread: update the row of a key by field name
  oGrid.Data.UpdateRow('MSFT', ['Price', 'Volume'], [413.1, 1350]);
end;
```

AppendRow recibe los valores en el orden de las columnas; UpdateRow recibe nombres de campo, de modo que un mensaje que solo lleva un precio solo actualiza el precio. UpdateRow también inserta una clave que todavía no existe, salvo que Data.AutoInsert sea False.

Consulte también

- Inicio rápido FMX
- Actualizaciones en vivo y subprocesos
- TsgcVCLGrid

Inicio rápido FMX

TsgcFMXGrid necesita Delphi XE6 o posterior. Colóquelo en un formulario de FireMonkey, o créelo desde código. Tiene las mismas columnas, datos y eventos que la cuadrícula VCL.

```
uses
  System.Classes, FMX.Forms, FMX.Types,
  sgcUICore_Theme, sgcGrid_Columns, sgcGrid_Data, sgcGrid_FMX;

procedure CreateGrid(aForm: TForm);
var
  oGrid: TsgcFMXGrid;
begin
  oGrid := TsgcFMXGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := TAlignLayout.Client;
  oGrid.Theme := utDark;

  oGrid.Columns.Add('Symbol', 'Symbol');
  oGrid.Columns.Add('Price', 'Price', gdtFloat).DisplayFormat := '0.00';
  oGrid.Columns.Add('Volume', 'Volume', gdtInteger);

  oGrid.Data.AppendRow('MSFT', ['MSFT', 412.5, 1200]);
  oGrid.Data.AppendRow('AAPL', ['AAPL', 228.1, 3400]);
  oGrid.Data.UpdateRow('MSFT', ['Price'], [413.1]);
end;
```

FireMonkey no tiene barras de desplazamiento nativas en un control de trama, por lo que la cuadrícula dibuja las suyas. La rueda del ratón desplaza tres filas, Mayús desplaza en horizontal y un gesto de arrastre desplaza con el dedo en las pantallas táctiles.

Consulte también

- Inicio rápido VCL
- TsgcFMXGrid

Inicio rápido C++Builder

C++Builder usa los mismos componentes mediante los encabezados y las bibliotecas estáticas que generan los paquetes de C++Builder. Añada `sgcUI\Source` y `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits) a las rutas de inclusión y de bibliotecas, algo que el instalador ya hace para el IDE. Los encabezados y las bibliotecas de sgcGrid y sgcUICore están en esa única carpeta. Incluya los encabezados sgcGrid_*.hpp y enlace sgcGridC13.lib y sgcUICoreC13.lib, o instale los paquetes de tiempo de diseño y coloque un TsgcVCLGrid en un formulario.
Las matrices abiertas de Delphi se pasan con la macro OPENARRAY. Las llamadas a AppendRow y UpdateRow también son seguras para subprocesos desde C++.

Consulte también

- Inicio rápido VCL
- Demos
- Solución de problemas

Demos

Las demos están en la carpeta Demos de la instalación: los proyectos Delphi en Demos\Delphi y los proyectos C++Builder en Demos\CBuilder. Cada carpeta de demo tiene un readme.html. Además, cada demo acepta dos modificadores que se usan para la verificación de la compilación: /test ejecuta sus comprobaciones y termina con el número de fallos, y /snapshot guarda la ventana en un archivo PNG.

Demo	Delphi	C++Builder	Muestra
LiveTicker	Sí	Sí	Unas 1.000 actualizaciones de precio por segundo desde un hilo de trabajo con UpdateRow, una vista ordenada por precio, cambios con destello, una regla de formato para los cambios negativos, alternadores de TailMode y de tema oscuro, y las actualizaciones y los fotogramas por segundo.
LiveTickerFMX	Sí	No	La demo LiveTicker con un TsgcFMXGrid.
DataSetGrid	Sí	Sí	Un TsgcGridSource_DataSet enlazado a un TClientDataSet de 200 registros con KeyField, AutoColumns y LiveUpdates, una selección que sigue al registro actual en ambos sentidos, y exportación e importación de JSON y CSV.
Editing	Sí	Sí	Edición en el lugar con una columna de lista de selección, una columna booleana, OnValidate, OnEditError y OnEdited. No está en la edición Community.
GridAndChart	Sí	Sí	Un panel con un mosaico de cuadrícula y un mosaico de gráfico; un TsgcChartDataSource_Table enlazado a la cuadrícula representa sus filas. Necesita sgcCharts y sgcDashboard.
AskYourGrid	Sí	Sí	TsgcGridAIQuery filtrando, ordenando y resaltando 200 acciones a partir de preguntas en lenguaje natural, con el analizador basado en reglas. Necesita sgcWebSockets.
LiveFeed	Sí	No	Un TsgcGridFeedSource_WebSocket alimentado por un servidor WebSocket en el mismo proceso. Necesita sgcWebSockets.

Notas sobre C++Builder

- Las demos de C++Builder se compilan con el compilador clásico de C++Builder 13 para Win32 y Win64 y enlazan sgcGridC13.lib y sgcUICoreC13.lib.
- DataSetGrid enlaza MIDAS de forma estática en Win32. C++Builder no tiene una biblioteca MIDAS estática para Win64, por lo que la compilación de 64 bits necesita midas.dll junto al ejecutable.
- GridAndChart también enlaza sgcChartsC13.lib y sgcDashboardC13.lib, y AskYourGrid enlaza sgcGridWSC13.lib y sgcWebSocketsC13.lib.

Consulte también

- Inicio rápido VCL
- Inicio rápido C++Builder
- Solución de problemas

TsgcVCLGrid

La cuadrícula de datos en vivo para VCL, en la unidad sgcGrid_VCL. Posee un TsgcGridData y un TsgcGridRenderer, dibuja con el lienzo de trama de sgcUICore e implementa IsgcUITable (delegado en Data) e IsgcGridDataHost, de modo que un origen de datos de gráfico, un origen de flujo o un origen DataSet pueden enlazarse al propio componente de cuadrícula.
Usa las barras de desplazamiento nativas de Windows: la barra vertical cuenta filas (líneas de visualización mientras las filas están agrupadas) y la horizontal píxeles lógicos, y la fila de encabezado permanece fija. Los miembros de edición solo existen cuando SGC_GRID_EDIT está definido.

Controladores de eventos de los ejemplos de características

Los ejemplos de las páginas de Características asignan los métodos de esta pequeña clase a los eventos.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

type
  TGridHandlers = class
  public
    procedure FilterRow(Sender: TObject; aRow: Integer; var aAccept: Boolean);
    procedure SelectionChange(Sender: TObject);
    procedure CellClick(Sender: TObject; aRow, aCol: Integer);
    procedure ColumnResized(Sender: TObject; aCol: Integer);
    procedure GetCellStyle(Sender: TObject; aRow, aCol: Integer;
      var aBack, aFore: TsgcUIColor; var aBold: Boolean);
    procedure GetValue(Sender: TObject; aRow, aCol: Integer;
      var aValue: Variant);
    procedure Validate(Sender: TObject; aRow, aCol: Integer;
      var aValue: Variant; var aAccept: Boolean);
    procedure Edited(Sender: TObject; aRow, aCol: Integer;
      const aOldValue, aNewValue: Variant);
    procedure EditError(Sender: TObject; aRow, aCol: Integer;
      const aText: string; var aMessage: string);
  end;
```

Propiedades

Propiedad	Tipo	Descripción
Columns	TsgcGridColumns	Las columnas de la cuadrícula. Consulte TsgcGridColumns.
Data	TsgcGridData	Las filas, la ordenación, el filtro y el seguimiento de cambios. Pública. También implementa IsgcUITable para la cuadrícula. Consulte TsgcGridData.
Renderer	TsgcGridRenderer	El renderizador con el que se dibuja la cuadrícula. Pública, de solo lectura.
Theme	TsgcUITheme	utLight (predeterminado), utDark o utCustom.
AccentColor	TsgcUIColor	El acento de la selección y del destello de cambio. 0 usa el acento del tema.

FontSize	Double	El tamaño del texto de las celdas.
HeaderHeight, RowHeight	Integer	La altura del encabezado (32) y de cada fila (28), en píxeles lógicos.
GridLinesH, GridLinesV	Boolean	Dibuja las líneas de cuadrícula horizontales y verticales. True de forma predeterminada.
AlternateRows	Boolean	Sombrea una de cada dos filas. True de forma predeterminada.
SelectionMode	TsgcGridSelectionMode	gsmRow (predeterminado) selecciona filas completas, gsmCell selecciona celdas individuales.
BorderStyle	TBorderStyle (VCL), TsgcGridBorderStyle (FMX)	Un marco simple con el color Border del tema, o ninguno.
MaxRows	Integer	Modo cola: 0 conserva todas las filas, un valor mayor que 0 elimina las filas más antiguas al insertar.
TailMode	Boolean	Mantiene la fila más reciente a la vista mientras la cuadrícula está desplazada hasta el final.
FilterText	string	Muestra solo las filas cuyo texto visible lo contiene, sin distinguir mayúsculas de minúsculas.
ReadOnly	Boolean	True de forma predeterminada. False habilita la edición en el lugar de las columnas Editable (SGC_GRID_EDIT).
FlashChanges, FlashDuration, FlashMode	Boolean, Integer, TsgcGridFlashMode	Hace destellar las celdas modificadas, durante cuántos milisegundos (800), en el fondo o en el texto.
FlashUpColor, FlashDownColor, FlashChangedColor	TsgcUIColor	Los colores del destello; 0 usa los predeterminados del tema.
FormatRules	TsgcGridFormatRules	Las reglas de formato condicional. Consulte Formato condicional.
MaxFPS	Integer	El mayor número de repintados por segundo (60).
FrameCount	Cardinal	El número de fotogramas dibujados hasta el momento. Pública, de solo lectura.
TopRow, ScrollX	Integer	La primera fila visible (la primera línea de visualización mientras las filas están agrupadas) y la posición de desplazamiento horizontal. Pública.
SelectedRow, SelectedCol	Integer	La fila de vista y el índice de columna seleccionados, -1 si no hay ninguno. Pública.
SelectedKey	string	La clave de la fila seleccionada. La selección sigue a su clave cuando cambia la vista. Pública, de solo lectura.
FrozenColumns	Integer	El número de columnas visibles que no se desplazan horizontalmente, 0 por defecto. Vea Columnas fijas.
ShowFooter	Boolean	Muestra la fila de pie con los resúmenes de las columnas, False por defecto. Vea Resúmenes.

ShowGroupPanel, GroupPanelHeight	Boolean, Integer	Muestra el panel de grupos sobre el encabezado (False por defecto) y su altura en DIP (36). Vea Agrupación.
GroupIndent	Integer	La sangría de cada nivel de grupo en DIP, 20 por defecto.
GroupsExpanded	Boolean	El estado de los grupos nuevos, True por defecto. Al asignarlo se conserva el estado de los grupos actuales.
GroupSummaries	Boolean	Muestra los resúmenes de las columnas en las filas de grupo, True por defecto.
Lines	TsgcGridLineView	Las líneas de visualización: filas de grupo y filas de la vista. Pública, de solo lectura. Vea TsgcGridLineView.
FocusedLine	Integer	La línea de visualización con el foco, -1 si no hay ninguna. En una línea de grupo SelectedRow es -1. Pública.
Tree	TsgcGridTreeOptions	Filas de árbol: KeyField, ParentField, TreeField, Indent y Expanded. El modo árbol está activo cuando KeyField y ParentField nombran columnas de la cuadrícula. Solo en la edición Registered. Vea Filas de árbol.
Detail	TsgcGridDetailOptions	Filas maestro detalle: Active, Source, MasterKeyField, DetailKeyField, Lines y Columns. Solo en la edición Registered. Vea Maestro detalle.

Métodos

Método	Devuelve	Descripción
ScrollToRow(aRow)		Desplaza la vista hasta mostrar la fila de vista.
RowAt(X, Y), ColAt(X, Y)	Integer	La fila de vista y el índice de columna situados bajo un punto, -1 si no hay ninguno.
SaveToPNG(aFileName, aWidth, aHeight)		Escribe la vista actual en un archivo PNG.
SaveToSVG(aFileName, aWidth, aHeight)		Escribe la vista actual en un archivo SVG.
StartEdit(aRow, aCol)	Boolean	Abre el editor en una celda, o alterna una celda booleana (SGC_GRID_EDIT).
CommitEdit	Boolean	Convierte, valida y escribe el texto del editor; False mantiene abierto el editor (SGC_GRID_EDIT).
CancelEdit		Cierra el editor sin escribir (SGC_GRID_EDIT).
EditorActive, Editor, EditRow, EditCol		El estado del editor abierto (SGC_GRID_EDIT). Propiedades.
GroupBy(aFieldNames), ClearGrouping		Agrupa por los campos indicados (FieldName, nivel 0 primero; los campos desconocidos se ignoran), o quita la agrupación.
ExpandAll, CollapseAll		Expande o contrae todos los grupos y los grupos nuevos a partir de ahora. En modo árbol actúan sobre los nodos del árbol (edición Registered).

ExpandGroup(aGroup), CollapseGroup(aGroup)		Expande o contrae un solo grupo.
GroupCount	Integer	El número de grupos.
GroupInfo(aGroup)	TsgcGridGroupInfo	Nivel, padre, columna, miembro, número de filas, estado y línea de un grupo.
FooterSummary(aCol), FooterSummaryText(aCol)	Variant, string	El valor de pie de una columna (índice de Columns) y su texto; Null y una cadena vacía cuando la columna no tiene SummaryType.
ExpandNode(aRow), CollapseNode(aRow)		Expande o contrae el nodo de árbol de la fila de la vista aRow y genera OnTreeExpandChange. Solo en la edición Registered.
ExpandDetail(aRow), CollapseDetail(aRow)		Abre o cierra la banda de detalle de la fila maestra de la vista aRow y genera OnDetailExpandChange. Solo en la edición Registered.
DetailExpanded(aRow)	Boolean	True cuando la banda de la fila de la vista aRow está abierta. Solo en la edición Registered.
DetailData(aRow)	TsgcGridData	Las filas de la banda de la fila de la vista aRow, nil cuando está contraída. Solo lectura. Solo en la edición Registered.
SaveToXLSX(aStream; aSheetName = 'Sheet1'), SaveToXLSX(aFileName; aSheetName = 'Sheet1')		Escribe la cuadrícula como un libro de Excel con los anchos de columna actuales y FrozenColumns. Solo en la edición Registered. Vea Exportación a Excel.

Eventos

Evento	Tipo	Descripción
OnCellClick, OnCellDblClick	TsgcGridCellEvent	Se hizo clic o doble clic en una celda: fila de vista e índice de columna.
OnSelectionChange	TNotifyEvent	Cambió la fila o la celda seleccionada.
OnSortChange	TNotifyEvent	Un clic en el encabezado cambió la ordenación.
OnColumnResized	TsgcGridColumnEvent	Se cambió el tamaño de una columna con el ratón.
OnGetCellStyle	TsgcGridCellStyleEvent	La última palabra sobre los colores y el estilo de negrita de cada celda dibujada. Consulte Formato condicional.
OnBeforeEdit	TsgcGridBeforeEditEvent	Un editor está a punto de abrirse; establezca aAllow en False para rechazarlo (SGC_GRID_EDIT).
OnValidate	TsgcGridValidateEvent	El valor convertido antes de escribirse; puede modificarlo o rechazarlo (SGC_GRID_EDIT).
OnEdited	TsgcGridEditedEvent	Se escribió un valor, con el valor anterior y el nuevo (SGC_GRID_EDIT).

OnEditError	TsgcGridEditErrorEvent	El texto no se convirtió al DataType de la columna; aMessage se puede modificar (SGC_GRID_EDIT).
OnGroupingChange	TNotifyEvent	Las columnas agrupadas o sus niveles han cambiado, desde el panel de grupos o desde código.
OnGroupExpandChange	TsgcGridGroupExpandEvent	Se expandió o contrajo un grupo; aGroup es -1 tras ExpandAll y CollapseAll.
OnGetGroupText	TsgcGridGetGroupTextEvent	Cambia el título aText de la fila de grupo aGroup. Se genera con el bloqueo de datos retenido.
OnTreeExpandChange	TsgcGridRowExpandEvent	Se expandió o contrajo un nodo del árbol; aRow es -1 tras ExpandAll y CollapseAll en modo árbol. Solo en la edición Registered.
OnDetailExpandChange	TsgcGridRowExpandEvent	Se abrió o cerró una banda de detalle; aRow es la fila maestra de la vista. Solo en la edición Registered.
OnGetDetailData	TsgcGridGetDetailDataEvent	Rellena las filas de una banda (aData) para la clave maestra aMasterKey; asigne True a aHandled para conservarlas. Sender es la cuadrícula. Se genera sin ningún bloqueo retenido. Solo en la edición Registered.
OnDetailCellClick	TsgcGridDetailCellEvent	Se hizo clic en una celda de una banda: aRow es la fila maestra de la vista, aDetailRow y aDetailCol la fila de la vista y el índice de Columns de los datos de la banda. Solo en la edición Registered.

Ejemplo

```
uses
  Forms, Classes, Controls,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_VCL;

procedure CreateGrid(aForm: TForm);
var
  oGrid: TsgcVCLGrid;
begin
  oGrid := TsgcVCLGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := alClient;

  // the columns: FieldName, Caption and DataType
  oGrid.Columns.Add('Symbol', 'Symbol');
  oGrid.Columns.Add('Price', 'Price', gdtFloat).DisplayFormat := '0.00';
  oGrid.Columns.Add('Volume', 'Volume', gdtInteger);

  // one row per key, the values in column order
  oGrid.Data.AppendRow('MSFT', ['MSFT', 412.5, 1200]);
  oGrid.Data.AppendRow('AAPL', ['AAPL', 228.1, 3400]);

  // later, from any thread: update the row of a key by field name
  oGrid.Data.UpdateRow('MSFT', ['Price', 'Volume'], [413.1, 1350]);
end;
```

Consulte también

- TsgcGridColumns
- TsgcGridData
- TsgcGridRenderer
- Selección y teclado
- Edición en el lugar

- Agrupación
- Resúmenes
- Columnas fijas
- TsgcGridLineView
- Filas de árbol
- Maestro detalle
- Exportación a Excel

TsgcFMXGrid

La cuadrícula de datos en vivo para FireMonkey, en la unidad sgcGrid_FMX (Delphi XE6 y posteriores, Windows, macOS, iOS y Android). Publica las propiedades, los métodos y los eventos de TsgcVCLGrid, además de los miembros habituales de FireMonkey, como Position, Size, Margins, Opacity y los eventos de ratón y de teclado.
La cuadrícula dibuja sus propias barras de desplazamiento en los bordes derecho e inferior mientras el contenido es mayor que la vista: el cursor se puede arrastrar y una pulsación sobre la pista avanza por páginas. La rueda del ratón desplaza tres filas (Mayús: en horizontal) y un gesto de arrastre desplaza con el dedo. Ctrl+C copia la selección, Cmd+C en macOS. BorderStyle es un TsgcGridBorderStyle, gbsNone o gbsSingle. Desde un hilo escritor, la cuadrícula pone en cola una actualización de la vista para el hilo principal con TThread.Queue.

Propiedades

Propiedad	Tipo	Descripción
Columns	TsgcGridColumns	Las columnas de la cuadrícula. Consulte TsgcGridColumns.
Data	TsgcGridData	Las filas, la ordenación, el filtro y el seguimiento de cambios. Pública. También implementa IsgcUITable para la cuadrícula. Consulte TsgcGridData.
Renderer	TsgcGridRenderer	El renderizador con el que se dibuja la cuadrícula. Pública, de solo lectura.
Theme	TsgcUITheme	utLight (predeterminado), utDark o utCustom.
AccentColor	TsgcUIColor	El acento de la selección y del destello de cambio. 0 usa el acento del tema.
FontSize	Double	El tamaño del texto de las celdas.
HeaderHeight, RowHeight	Integer	La altura del encabezado (32) y de cada fila (28), en píxeles lógicos.
GridLinesH, GridLinesV	Boolean	Dibuja las líneas de cuadrícula horizontales y verticales. True de forma predeterminada.
AlternateRows	Boolean	Sombrea una de cada dos filas. True de forma predeterminada.
SelectionMode	TsgcGridSelectionMode	gsmRow (predeterminado) selecciona filas completas, gsmCell selecciona celdas individuales.
BorderStyle	TBorderStyle (VCL), TsgcGridBorderStyle (FMX)	Un marco simple con el color Border del tema, o ninguno.
MaxRows	Integer	Modo cola: 0 conserva todas las filas, un valor mayor que 0 elimina las filas más antiguas al insertar.
TailMode	Boolean	Mantiene la fila más reciente a la vista mientras la cuadrícula está desplazada hasta el final.
FilterText	string	Muestra solo las filas cuyo texto visible lo contiene, sin distinguir mayúsculas de minúsculas.
ReadOnly	Boolean	True de forma predeterminada. False habilita la edición en el lugar de las columnas Editable (SGC_GRID_EDIT).

FlashChanges, FlashDuration, FlashMode	Boolean, Integer, TsgcGridFlashMode	Hace destellar las celdas modificadas, durante cuántos milisegundos (800), en el fondo o en el texto.
FlashUpColor, FlashDownColor, FlashChangedColor	TsgcUIColor	Los colores del destello; 0 usa los predeterminados del tema.
FormatRules	TsgcGridFormatRules	Las reglas de formato condicional. Consulte Formato condicional.
MaxFPS	Integer	El mayor número de repintados por segundo (60).
FrameCount	Cardinal	El número de fotogramas dibujados hasta el momento. Pública, de solo lectura.
TopRow, ScrollX	Integer	La primera fila visible (la primera línea de visualización mientras las filas están agrupadas) y la posición de desplazamiento horizontal. Pública.
SelectedRow, SelectedCol	Integer	La fila de vista y el índice de columna seleccionados, -1 si no hay ninguno. Pública.
SelectedKey	string	La clave de la fila seleccionada. La selección sigue a su clave cuando cambia la vista. Pública, de solo lectura.
FrozenColumns	Integer	El número de columnas visibles que no se desplazan horizontalmente, 0 por defecto. Vea Columnas fijas.
ShowFooter	Boolean	Muestra la fila de pie con los resúmenes de las columnas, False por defecto. Vea Resúmenes.
ShowGroupPanel, GroupPanelHeight	Boolean, Integer	Muestra el panel de grupos sobre el encabezado (False por defecto) y su altura en DIP (36). Vea Agrupación.
GroupIndent	Integer	La sangría de cada nivel de grupo en DIP, 20 por defecto.
GroupsExpanded	Boolean	El estado de los grupos nuevos, True por defecto. Al asignarlo se conserva el estado de los grupos actuales.
GroupSummaries	Boolean	Muestra los resúmenes de las columnas en las filas de grupo, True por defecto.
Lines	TsgcGridLineView	Las líneas de visualización: filas de grupo y filas de la vista. Pública, de solo lectura. Vea TsgcGridLineView.
FocusedLine	Integer	La línea de visualización con el foco, -1 si no hay ninguna. En una línea de grupo SelectedRow es -1. Pública.
Tree	TsgcGridTreeOptions	Filas de árbol: KeyField, ParentField, TreeField, Indent y Expanded. El modo árbol está activo cuando KeyField y ParentField nombran columnas de la cuadrícula. Solo en la edición Registered. Vea Filas de árbol.
Detail	TsgcGridDetailOptions	Filas maestro detalle: Active, Source, MasterKeyField, DetailKeyField, Lines y Columns. Solo en la edición Registered. Vea Maestro detalle.

Métodos

Método	Devuelve	Descripción
ScrollToRow(aRow)		Desplaza la vista hasta mostrar la fila de vista.
RowAt(X, Y), ColAt(X, Y)	Integer	La fila de vista y el índice de columna situados bajo un punto, -1 si no hay ninguno.
SaveToPNG(aFileName, aWidth, aHeight)		Escribe la vista actual en un archivo PNG.
SaveToSVG(aFileName, aWidth, aHeight)		Escribe la vista actual en un archivo SVG.
StartEdit(aRow, aCol)	Boolean	Abre el editor en una celda, o alterna una celda booleana (SGC_GRID_EDIT).
CommitEdit	Boolean	Convierte, valida y escribe el texto del editor; False mantiene abierto el editor (SGC_GRID_EDIT).
CancelEdit		Cierra el editor sin escribir (SGC_GRID_EDIT).
EditorActive, Editor, EditRow, EditCol		El estado del editor abierto (SGC_GRID_EDIT). Propiedades.
GroupBy(aFieldNames), ClearGrouping		Agrupa por los campos indicados (FieldName, nivel 0 primero; los campos desconocidos se ignoran), o quita la agrupación.
ExpandAll, CollapseAll		Expande o contrae todos los grupos y los grupos nuevos a partir de ahora. En modo árbol actúan sobre los nodos del árbol (edición Registered).
ExpandGroup(aGroup), CollapseGroup(aGroup)		Expande o contrae un solo grupo.
GroupCount	Integer	El número de grupos.
GroupInfo(aGroup)	TsgcGridGroupInfo	Nivel, padre, columna, miembro, número de filas, estado y línea de un grupo.
FooterSummary(aCol), FooterSummaryText(aCol)	Variant, string	El valor de pie de una columna (índice de Columns) y su texto; Null y una cadena vacía cuando la columna no tiene SummaryType.
ExpandNode(aRow), CollapseNode(aRow)		Expande o contrae el nodo de árbol de la fila de la vista aRow y genera OnTreeExpandChange. Solo en la edición Registered.
ExpandDetail(aRow), CollapseDetail(aRow)		Abre o cierra la banda de detalle de la fila maestra de la vista aRow y genera OnDetailExpandChange. Solo en la edición Registered.
DetailExpanded(aRow)	Boolean	True cuando la banda de la fila de la vista aRow está abierta. Solo en la edición Registered.

DetailData(aRow)	TsgcGridData	Las filas de la banda de la fila de la vista aRow, nil cuando está contraída. Solo lectura. Solo en la edición Registered.
SaveToXLSX(aStream; aSheetName = 'Sheet1'), SaveToXLSX(aFileName; aSheetName = 'Sheet1')		Escribe la cuadrícula como un libro de Excel con los anchos de columna actuales y FrozenColumns. Solo en la edición Registered. Vea Exportación a Excel.

Eventos

Evento	Tipo	Descripción
OnCellClick, OnCellDblClick	TsgcGridCellEvent	Se hizo clic o doble clic en una celda: fila de vista e índice de columna.
OnSelectionChange	TNotifyEvent	Cambió la fila o la celda seleccionada.
OnSortChange	TNotifyEvent	Un clic en el encabezado cambió la ordenación.
OnColumnResized	TsgcGridColumnEvent	Se cambió el tamaño de una columna con el ratón.
OnGetCellStyle	TsgcGridCellStyleEvent	La última palabra sobre los colores y el estilo de negrita de cada celda dibujada. Consulte Formato condicional.
OnBeforeEdit	TsgcGridBeforeEditEvent	Un editor está a punto de abrirse; establezca aAllow en False para rechazarlo (SGC_GRID_EDIT).
OnValidate	TsgcGridValidateEvent	El valor convertido antes de escribirse; puede modificarlo o rechazarlo (SGC_GRID_EDIT).
OnEdited	TsgcGridEditedEvent	Se escribió un valor, con el valor anterior y el nuevo (SGC_GRID_EDIT).
OnEditError	TsgcGridEditErrorEvent	El texto no se convirtió al DataType de la columna; aMessage se puede modificar (SGC_GRID_EDIT).
OnGroupingChange	TNotifyEvent	Las columnas agrupadas o sus niveles han cambiado, desde el panel de grupos o desde código.
OnGroupExpandChange	TsgcFMXGridGroupExpandEvent	Se expandió o contrajo un grupo; aGroup es -1 tras ExpandAll y CollapseAll.
OnGetGroupText	TsgcGridGetGroupTextEvent	Cambia el título aText de la fila de grupo aGroup. Se genera con el bloqueo de datos retenido.
OnTreeExpandChange	TsgcFMXGridRowExpandEvent	Se expandió o contrajo un nodo del árbol; aRow es -1 tras ExpandAll y CollapseAll en modo árbol. Solo en la edición Registered.
OnDetailExpandChange	TsgcFMXGridRowExpandEvent	Se abrió o cerró una banda de detalle; aRow es la fila maestra de la vista. Solo en la edición Registered.
OnGetDetailData	TsgcGridGetDetailDataEvent	Rellena las filas de una banda (aData) para la clave maestra aMasterKey; asigne True a aHandled para conservarlas. Sender es la cuadrícula. Se genera sin ningún bloqueo retenido. Solo en la edición Registered.

OnDetailCellClick	TsgcFMXGridDetailCellEvent	Se hizo clic en una celda de una banda: aRow es la fila maestra de la vista, aDetailRow y aDetailCol la fila de la vista y el índice de Columns de los datos de la banda. Solo en la edición Registered.

Ejemplo

```
uses
  System.Classes, FMX.Forms, FMX.Types,
  sgcUICore_Theme, sgcGrid_Columns, sgcGrid_Data, sgcGrid_FMX;

procedure CreateGrid(aForm: TForm);
var
  oGrid: TsgcFMXGrid;
begin
  oGrid := TsgcFMXGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := TAlignLayout.Client;
  oGrid.Theme := utDark;

  oGrid.Columns.Add('Symbol', 'Symbol');
  oGrid.Columns.Add('Price', 'Price', gdtFloat).DisplayFormat := '0.00';
  oGrid.Columns.Add('Volume', 'Volume', gdtInteger);

  oGrid.Data.AppendRow('MSFT', ['MSFT', 412.5, 1200]);
  oGrid.Data.AppendRow('AAPL', ['AAPL', 228.1, 3400]);
  oGrid.Data.UpdateRow('MSFT', ['Price'], [413.1]);
end;
```

Consulte también

- TsgcGridColumns
- TsgcGridData
- TsgcGridRenderer
- Selección y teclado
- Edición en el lugar
- Agrupación
- Resúmenes
- Columnas fijas
- TsgcGridLineView
- Filas de árbol
- Maestro detalle
- Exportación a Excel

TsgcGridColumn

Una columna de una cuadrícula, un elemento de TsgcGridColumns, en la unidad sgcGrid_Columns. Los datos direccionan los valores por el ID de la columna, de modo que mover o eliminar una columna nunca desplaza los valores de las demás.

Propiedades

Propiedad	Tipo	Descripción
FieldName	string	El nombre que usan para la columna los datos, los flujos, las importaciones y las reglas de formato.
Caption	string	El texto del encabezado.
DataType	TsgcGridDataType	gdtText (predeterminado), gdtInteger, gdtFloat, gdtCurrency, gdtDateTime o gdtBoolean. Determina la ordenación, el formato, la alineación, la edición y las comparaciones de las reglas.
Width, MinWidth	Integer	El ancho (100) y el ancho mínimo que permite un cambio de tamaño con el ratón (20), en píxeles lógicos.
Alignment	TsgcGridAlignment	gaLeft, gaCenter, gaRight o gaAuto (predeterminado): números a la derecha, texto a la izquierda, booleanos centrados.
DisplayFormat	string	Un formato FormatFloat para números o un formato FormatDateTime para fechas. Una columna booleana sin DisplayFormat se dibuja como una marca de verificación.
Visible	Boolean	Muestra la columna. Las columnas ocultas conservan sus valores y su índice.
Editable	Boolean	La columna se puede editar en el lugar cuando la cuadrícula no es ReadOnly. True de forma predeterminada.
SortOrder, SortIndex	TsgcGridSortOrder, Integer	gsoNone, gsoAscending o gsoDescending, y la posición de la columna en una ordenación por varias columnas.
PickList	TStrings	Los valores del editor de cuadro combinado de la columna.
Tag	Integer	Libre para la aplicación.
GroupIndex	Integer	0 o más agrupa la cuadrícula por esta columna en ese nivel (niveles ordenados por GroupIndex y después por Index); -1 (por defecto) no agrupa. Vea Agrupación.
SummaryType	TsgcGridSummaryType	gsuNone (por defecto), gsuSum, gsuCount, gsuAverage, gsuMin o gsuMax: el resumen de la columna en el pie y en las filas de grupo. Vea Resúmenes.
SummaryFormat	string	La máscara de FormatFloat o FormatDateTime del resumen, texto literal entre comillas dobles; vacío usa DisplayFormat, y 0 para gsuCount.

Métodos

Método	Devuelve	Descripción
IsNumeric	Boolean	True para gdtInteger, gdtFloat y gdtCurrency.
EffectiveAlignment	TsgcGridAlignment	La alineación en la que se resuelve gaAuto.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupColumns(Grid: TsgcVCLGrid);
var
  oColumn: TsgcGridColumn;
begin
  Grid.Columns.Clear;
  Grid.Columns.Add('Symbol', 'Symbol');
  oColumn := Grid.Columns.Add('Price', 'Price', gdtFloat);
  oColumn.DisplayFormat := '#,##0.00';
  oColumn.Width := 90;
  Grid.Columns.Add('Change', 'Change %', gdtFloat).DisplayFormat :=
    '+0.00;-0.00;0.00';
  Grid.Columns.Add('Volume', 'Volume', gdtInteger).DisplayFormat := '#,##0';
  Grid.Columns.Add('Time', 'Updated', gdtDateTime).DisplayFormat := 'hh:nn:ss';
  Grid.Columns.Add('Halted', 'Halted', gdtBoolean);
end;
```

Consulte también

- TsgcGridColumns
- Ordenación
- Edición en el lugar
- Agrupación
- Resúmenes

TsgcGridColumns

La colección de elementos TsgcGridColumn de una cuadrícula o de un TsgcGridData independiente, en la unidad sgcGrid_Columns.

Propiedades

Propiedad	Tipo	Descripción
Items[Index]	TsgcGridColumn	Las columnas, propiedad predeterminada.
VisibleColumn[Index]	TsgcGridColumn	Las columnas visibles en orden.
Version	Cardinal	Se incrementa con cada cambio de las columnas.

Métodos

Método	Devuelve	Descripción
Add	TsgcGridColumn	Añade una columna vacía.
Add(aFieldName, aCaption, aDataType)	TsgcGridColumn	Añade una columna con un nombre de campo, un título y un tipo de datos (gdtText de forma predeterminada).
FindByField(aFieldName)	TsgcGridColumn	La columna de un nombre de campo, nil si no existe.
IndexOfField(aFieldName)	Integer	El índice de la columna de un nombre de campo, -1 si no existe.
VisibleCount	Integer	El número de columnas visibles.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Cambió cualquier propiedad de una columna.
OnModelChange	TNotifyEvent	Un cambio que afecta a los datos o a la vista, como una ordenación o una columna nueva.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SortByChange(Grid: TsgcVCLGrid);
begin
```

```
  Grid.Data.BeginUpdate;
  try
    // first by Change, highest first, then by Symbol
    Grid.Columns.FindByField('Change').SortIndex := 0;
    Grid.Columns.FindByField('Change').SortOrder := gsoDescending;
    Grid.Columns.FindByField('Symbol').SortIndex := 1;
    Grid.Columns.FindByField('Symbol').SortOrder := gsoAscending;
  finally
    Grid.Data.EndUpdate;
  end;
end;
```

Consulte también

- TsgcGridColumn
- TsgcGridData

TsgcGridData

El almacén de filas de sgcGrid, independiente del framework y seguro para subprocesos, en la unidad sgcGrid_Data. Cada cuadrícula posee uno en su propiedad Data, y se puede usar por sí solo, por ejemplo con TsgcGridRenderer. Create(nil) crea y posee sus propias columnas.
Las filas se conservan en orden de inserción y se localizan por clave en O(log n). Los índices de fila de Cells, CellText, KeyOf, RowOfKey y CellChange son filas de vista (ordenadas y filtradas); los índices de columna son índices de Columns, incluidas las columnas ocultas. Una lectura fuera de rango devuelve Null, una cadena vacía o cdNone y nunca lanza una excepción, porque un hilo escritor puede eliminar una fila entre un recuento y una lectura.

Propiedades

Propiedad	Tipo	Descripción
Columns	TsgcGridColumns	Las columnas.
Cells[aRow, aCol]	Variant	El valor de una fila de vista.
CellText[aRow, aCol]	string	El valor formateado con el DisplayFormat de la columna.
StoreCells[aStoreRow, aCol]	Variant	El valor de una fila del almacén, para OnFilterRow.
FormatSettings	TFormatSettings	Los ajustes usados para formatear y analizar valores.
AutoInsert	Boolean	UpdateRow y UpdateCell insertan una clave que falta. True de forma predeterminada.
MaxRows	Integer	0 sin límite, en otro caso se eliminan las filas más antiguas al insertar.
FilterText	string	Conserva las filas cuyo texto visible lo contiene, sin distinguir mayúsculas de minúsculas.
Virtual, VirtualRowCount	Boolean, Integer	El modo virtual y su número de filas. Consulte Modo virtual.
LayoutVersion	Cardinal	Se incrementa cuando la disposición de agrupación o de árbol de las filas de la vista puede haber cambiado: insertar, eliminar, Clear, cualquier cambio de vista o de columna, un cambio de las columnas agrupadas y un cambio de valor en una columna con GroupIndex 0 o más o en un campo pasado a SetLayoutFields. Un cambio de valor en cualquier otra columna no lo modifica. La vista de líneas de las cuadrículas lo lee. Solo lectura.

Métodos

Método	Devuelve	Descripción
AppendRow(aKey, aValues)	Integer	Añade una fila con los valores en el orden de las columnas y devuelve su fila del almacén. Una clave vacía genera una clave única #<n>; una clave existente actualiza esa fila.

UpdateRow(aKey, aFields, aValues)		Establece los campos indicados de una fila. Los campos desconocidos se ignoran.
UpdateCell(aKey, aField, aValue)		Establece un campo de una fila.
DeleteRow(aKey)	Boolean	Elimina la fila de una clave.
Clear		Elimina todas las filas.
BeginUpdate, EndUpdate, IsUpdating		Agrupa los cambios en un único OnChange.
Lock, Unlock		Mantiene el bloqueo interno durante varias lecturas.
RowCount, ViewCount	Integer	Las filas almacenadas (VirtualRowCount en modo virtual) y las filas que pasan el filtro.
ViewRow(aRow)	Integer	La fila del almacén de una fila de vista, -1 si está fuera de rango.
KeyOf(aRow), RowOfKey(aKey), StoreKeyOf(aStoreRow)		Convierten entre filas de vista, filas del almacén y claves. RowOfKey devuelve -1 cuando la clave no existe o está filtrada.
CellChange(aRow, aCol)	TsgcGridCellChange	La Direction (cdNone, cdUp, cdDown, cdChanged) y la Age en milisegundos del último cambio de una celda.
GetTick	Int64	El reloj monótono en milisegundos de las marcas de cambio.
InvalidateView		Reconstruye la vista, por ejemplo después de cambiar la lógica de OnFilterRow.
UseInvariantFormat		Establece FormatSettings con separador decimal punto, separador de millares coma, yyyy-mm-dd y hh:nn:ss.
SetLayoutFields(aFields)		Campos (FieldName, sin distinguir mayúsculas) cuyos cambios de valor también mueven LayoutVersion, como las columnas agrupadas; una matriz vacía borra la lista.

Eventos

Evento	Tipo	Descripción
OnChange	TsgcGridChangeEvent	gckData, gckView o gckStructure, lanzado tras liberar el bloqueo en el hilo escritor.
OnFilterRow	TsgcGridFilterRowEvent	Acepta o rechaza una fila del almacén. Se llama con el bloqueo retenido.
OnGetValue	TsgcGridGetValueEvent	Los valores del modo virtual.
OnGetTick	TsgcGridGetTickEvent	Sustituye el reloj de las marcas de cambio, por ejemplo en las pruebas.

TsgcGridData también implementa IsgcUITable, consulte IsgcGridDataHost e IsgcUITable.

Ejemplo

```
uses
  SysUtils,
  sgcGrid_Columns, sgcGrid_Data;

    oData.Columns[1].SortOrder := gsoAscending;
    oData.Lock;
    try
      // view rows: sorted and filtered
      for i := 0 to oData.ViewCount - 1 do
        WriteLn(oData.KeyOf(i), ' = ', oData.CellText[i, 1]);
    finally
      oData.Unlock;
    end;
```

Consulte también

- Actualizaciones en vivo y subprocesos
- Filtrado
- Ordenación
- TsgcGridLineView

TsgcGridLineView

La vista de líneas de sgcGrid, independiente del framework, en la unidad sgcGrid_Lines. Asigna las líneas de visualización de una cuadrícula a las filas de la vista de su TsgcGridData. Cada cuadrícula tiene una en su propiedad Lines, y puede usarse por sí sola con un TsgcGridData y el TsgcGridRenderer. No es propietaria de los datos: libérela antes que los datos.
Una vista de líneas es estado de la vista. Los datos, sus filas de la vista e IsgcUITable no cambian. Cada línea mide una fila de alto, de modo que un índice de línea es una posición de desplazamiento. Sin ninguna columna agrupada (y sin árbol ni banda de detalle expandida en la edición Registered), o con unos datos virtuales, no se construye nada y la línea i es la fila de la vista i: es la identidad, la cuadrícula plana.

Tipos de línea

TsgcGridLineKind	Line
glkRow	Una fila de la vista. Row la devuelve.
glkGroup	Una fila de grupo. Group devuelve el grupo, Row devuelve -1.
glkDetail	Una línea de una banda de detalle (edición Registered, vea Maestro detalle). Row devuelve la fila maestra.

Las líneas agrupadas son las filas de grupo en orden de profundidad, cada una seguida de sus grupos hijos, y un grupo del último nivel, de sus filas de la vista. Las líneas de grupo tienen Level 0 en el nivel más externo, las líneas de fila Level = GroupLevels.

Métodos

Método	Returns	Descripción
Create(aData)		Crea una vista de líneas sobre aData.
Update		Actualiza las líneas con los datos. Cada lector de abajo lo llama, de modo que no cuesta nada cuando no ha cambiado nada.
Invalidate		Fuerza una reconstrucción en la lectura siguiente.
Identity	Boolean	True cuando la línea i es la fila de la vista i.
Count	Integer	El número de líneas de visualización.
Kind(aLine)	TsgcGrid-LineKind	El tipo de una línea, glkRow fuera de rango.
Row(aLine)	Integer	La fila de la vista de una línea de fila, la fila maestra de una línea de detalle, -1 para una línea de grupo o fuera de rango.
Level(aLine)	Integer	El nivel de sangría de una línea.
Group(aLine)	Integer	El grupo de una línea de grupo, el grupo más interno de una línea de fila, -1 en identidad o fuera de rango.
Expandable(aLine), Expanded(aLine)	Boolean	Si la línea puede expandirse o contraerse, y su estado.

ToggleLine(aLine)	Boolean	Alterna la línea; False cuando no tiene nada que alternar.
LineOfRow(aRow)	Integer	La línea de una fila de la vista, -1 dentro de un grupo contraído o fuera de rango.
GroupLevels, GroupCount	Integer	El número de niveles agrupados y de grupos.
GroupInfo(aGroup)	TsgcGrid-GroupInfo	Un grupo, de `0` a `GroupCount - 1` en orden de profundidad.
GroupSummary(aGroup, aCol)	Variant	El resumen de una columna (índice de Columns) sobre las filas del grupo; Null cuando la columna no tiene SummaryType o aGroup está fuera de rango.
GroupBy(aFieldNames), ClearGrouping		Asignan GroupIndex a las columnas indicadas (nivel 0 primero) y -1 a las demás, o -1 a todas las columnas.
SetGroupExpanded(aGroup, aExpanded)		Expande o contrae un solo grupo.
ExpandAll, CollapseAll		Todos los grupos, y los grupos nuevos a partir de ahora.
FlattenAll	TsgcGrid-FlatLines	Cada fila de grupo y cada fila de la vista en orden de línea, esté lo que esté contraído; Hidden marca las líneas dentro de un grupo contraído. Las líneas de detalle no se incluyen.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcGrid-Data	Los datos de la vista de líneas. Solo lectura.
GroupsExpanded	Boolean	El estado de los grupos que no se alternaron y de los grupos nuevos. Al asignarlo se conserva el estado de los grupos actuales.

Los miembros de árbol y de detalle (TreeMode, NodeExpanded, SetNodeExpanded, DetailExpandable, DetailExpanded, SetDetailExpanded, ToggleDetail, DetailView, DetailIndex, TreeOptions, DetailOptions, Tree y Detail) solo existen en la edición Registered, vea Filas de árbol y Maestro detalle.

TsgcGridGroupInfo

Campo	Tipo	Descripción
Level	Integer	0 para el nivel más externo.
Parent	Integer	El grupo padre, -1 en el nivel 0.
Column	Integer	El índice de Columns de la columna agrupada.
RowCount	Integer	Las filas de la vista del grupo, incluidos sus grupos hijos.
FirstRow	Integer	La fila de la vista de la primera fila del grupo, en orden de línea.
Text	string	El texto del miembro, (blank) para el miembro vacío.
Value	Variant	El valor tipado de la primera fila, Null para el miembro vacío.

Blank, Expanded	Boolean	Si el grupo es el miembro vacío, y su estado.
Line	Integer	La línea de la fila de grupo, -1 dentro de un grupo contraído.
Path	string	Los textos de los miembros desde el nivel 0, unidos con #1.

TsgcGridFlatLines es un registro con Count y las matrices Kind, Ref (la fila de la vista o el grupo), Level y Hidden.

Hilos

Llame a la vista de líneas desde el hilo que es propietario de la cuadrícula, el hilo principal en un control. Los hilos de escritura siguen escribiendo en los datos; la vista de líneas se reconstruye en su lectura siguiente cuando el LayoutVersion de los datos se ha movido, y solo actualiza los resúmenes de grupo cuando únicamente han cambiado valores. TsgcGridGrouper (unidad sgcGrid_Group) hace la agrupación por ella.

Ejemplo

Agrupa tres filas por Region sin ningún control e imprime cada grupo con su número de filas y la suma de Amount, y después los productos que contiene.

```
uses
  SysUtils, Classes, Variants,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_Summary, sgcGrid_Lines;

procedure PrintGroups(aOut: TStrings);
var
  oData: TsgcGridData;
  oLines: TsgcGridLineView;
  vAmount: TsgcGridColumn;
  vInfo: TsgcGridGroupInfo;
  i: Integer;
begin
  oData := TsgcGridData.Create(nil);
  try
    oData.Columns.Add('Region', 'Region');
    oData.Columns.Add('Product', 'Product');
    vAmount := oData.Columns.Add('Amount', 'Amount', gdtFloat);
    vAmount.SummaryType := gsuSum;
    oData.AppendRow('1', ['North', 'Apples', 120.0]);
    oData.AppendRow('2', ['South', 'Pears', 80.0]);
    oData.AppendRow('3', ['North', 'Pears', 45.5]);

    oLines := TsgcGridLineView.Create(oData);
    try
      oLines.GroupBy(['Region']);
      for i := 0 to oLines.Count - 1 do
        case oLines.Kind(i) of
          glkGroup:
            begin
              vInfo := oLines.GroupInfo(oLines.Group(i));
              aOut.Add(Format('%s: %d rows, %s', [vInfo.Text, vInfo.RowCount,
                sgcGridSummaryText(vAmount, gsuSum,
                oLines.GroupSummary(oLines.Group(i), vAmount.Index),
                oData.FormatSettings)]));
            end;
          glkRow:
            aOut.Add('  ' + oData.CellText[oLines.Row(i), 1]);
        end;
    finally
      oLines.Free;
    end;
  finally
    oData.Free;
```

```
  end;
end;
```

Consulte también

- Agrupación
- Resúmenes
- TsgcGridData
- TsgcGridRenderer
- Agrupación y resúmenes de .NET

TsgcGridDataStore

Un componente no visual, en la unidad sgcGrid_Detail, que guarda filas sin mostrarlas. Su misión es ser el Source de las bandas de detalle de una cuadrícula, vea Maestro detalle. Lo rellena un TsgcGridSource_DataSet cuyo Grid es el almacén, o se rellena por código, de modo que un detalle de TDataSet funciona desde el diseñador de formularios sin código. Está en la página SGC Grid de la paleta de componentes.
Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.
TsgcGridDataStore desciende de TComponent e implementa IsgcGridDataHost e IsgcUITable, esta última delegada en Data.

Propiedades

Propiedad	Tipo	Descripción
Columns	TsgcGridColumns	Las columnas de las filas. Publicada, de modo que se asigna en el diseñador. Con un TsgcGridSource_DataSet y AutoColumns True se crean a partir de los campos.
Data	TsgcGridData	Las filas, leídas y escritas como el Data de una cuadrícula: AppendRow, UpdateRow, DeleteRow, BeginUpdate y EndUpdate. Pública, de solo lectura.

Uso

- Asigne el almacén a Detail.Source de una cuadrícula, y a DetailKeyField la columna del almacén que contiene la clave maestra.
- Los cambios en las filas del almacén, por código o por el origen TDataSet, rellenan de nuevo las bandas de detalle abiertas en el fotograma siguiente.
- Liberar el almacén asigna nil a Detail.Source de la cuadrícula.

Ejemplo

Un TDataSet de pedidos cargado en un almacén, mostrado como el detalle de una cuadrícula de clientes:

```
uses
  SysUtils, Classes, DB,
  sgcGrid_Detail, sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure BindOrders(Grid: TsgcVCLGrid; Orders: TDataSet;
  Store: TsgcGridDataStore; Source: TsgcGridSource_DataSet);
begin
  // the TDataSet rows go into the store, not into a visible grid
  Source.DataSet := Orders;
  Source.Grid := Store;
  Source.KeyField := 'OrderID';
  // AutoColumns (True by default) creates the store columns from the fields
  Source.Active := True;
  // every expanded customer shows its orders
  Grid.Detail.Source := Store;
  Grid.Detail.MasterKeyField := 'CustomerID';
  Grid.Detail.DetailKeyField := 'CustomerID';
  Grid.Detail.Active := True;
end;
```

Lo mismo desde el diseñador: asigne DataSet, Grid (el almacén), KeyField y Active en el TsgcGridSource_DataSet, y después Detail.Source, Detail.MasterKeyField, Detail.DetailKeyField y Detail.Active en la cuadrícula. La demo TreeAndDetail funciona así.

Consulte también

- Maestro detalle
- TsgcGridData
- TsgcGridColumns
- TsgcGridSource_DataSet
- Ediciones
- Filas de árbol y maestro detalle de .NET

TsgcGridRenderer

El renderizador de sgcGrid, independiente del framework, en la unidad sgcGrid_Render. Dibuja un TsgcGridData en cualquier lienzo de sgcUICore, de trama o SVG, de modo que la cuadrícula VCL, la cuadrícula FMX y la salida sin interfaz tienen el mismo aspecto. La geometría está en píxeles lógicos; Render recibe el tamaño del lienzo en píxeles de dispositivo y la escala, y ajusta cada borde a píxeles de dispositivo enteros.

TsgcGridRenderOptions

El aspecto publicado de una cuadrícula: HeaderHeight, RowHeight, CellPadding (8), GridLinesH, GridLinesV, AlternateRows, SelectionMode, Theme, AccentColor, FontSize, HeaderFontSize, ShowSortGlyphs, FlashChanges, FlashDuration, FlashUpColor, FlashDownColor, FlashChangedColor, FlashMode y FormatRules, y la disposición de la vista de líneas FrozenColumns, ShowFooter, ShowGroupPanel, GroupPanelHeight, GroupIndent y GroupSummaries. Los controles de cuadrícula publican las mismas propiedades. Colors devuelve los colores resueltos del tema.

TsgcGridRenderState

Un registro con el estado de vista de un fotograma: ScrollX, ScrollY, SelectedRow, SelectedCol, HotRow, HotCol, FocusRow, FocusCol, Focused, FocusLine, HotLine y ColWidths. sgcGridRenderState devuelve un estado vacío sin nada seleccionado. FocusLine y HotLine son líneas de visualización, -1 por defecto; FocusLine -1 da el foco a la línea de FocusRow.

Propiedades

Propiedad	Tipo	Descripción
Options	TsgcGridRenderOptions	El aspecto.
CurrentData, CurrentNow	TsgcGridData, Int64	Los datos y el instante del fotograma que se está dibujando, válidos en OnGetCellStyle.
Fading	Boolean	True tras un fotograma que dibujó un destello todavía en curso.
FooterSummaries	TsgcGridSummaryCache	Los valores de pie, actualizados en cada fotograma con ShowFooter.

Métodos

Método	Devuelve	Descripción
Render(aData, aState, aCanvas, aWidth, aHeight, aScale, aNow)		Dibuja un fotograma.
SaveToPNG, SaveToSVG		Escriben un fotograma en un flujo o en un archivo, con una escala opcional.
HitTest(aData, aState, aX, aY)	TsgcGridHitInfo	La parte (ghpNone, ghpHeader, ghpHeaderResize, ghpCell), la fila y la columna situadas bajo un punto.

ColumnWidth, ColumnLeft, ColumnRight, ContentWidth, ContentHeight, VisibleRowRange		Funciones auxiliares de geometría.
Render(aData, aLines, aState, aCanvas, aWidth, aHeight, aScale, aNow)		Pinta un fotograma de líneas de visualización: el panel de grupos, el encabezado, las líneas y el pie. aLines nil es la cuadrícula plana. Vea TsgcGridLineView.
SaveToPNG(aData, aLines, ...), SaveToSVG(aData, aLines, ...)		Escriben un fotograma de líneas de visualización en un flujo o un archivo.
HitTest(aData, aLines, aState, aX, aY, aHeight)	TsgcGridHitInfo	Prueba de impacto sobre líneas de visualización. Pase la altura de la vista en aHeight para acertar en el pie; con 0 el pie nunca se acierta.
VisibleLineRange, GutterWidth, FrozenWidth, ContentWidth(aColumns, aLines, aState)		Geometría de la vista de líneas: las líneas visibles sin el panel de grupos ni el pie, el ancho de los chevrons de grupo, el ancho que no se desplaza y el ancho completo.
GroupPanelDropIndex, GroupChipBounds, PaintDragChip		La posición de destino en el panel de grupos, los límites de un chip y el dibujo del chip arrastrado, para los hosts que arrastran encabezados al panel de grupos.
GroupText(aLines, aGroup)	string	El título de una fila de grupo, con OnGetGroupText aplicado.

Sobre líneas, TsgcGridHitPart añade ghpGroupPanel, ghpGroupChip, ghpGroupRow, ghpExpandButton (el chevron), ghpFooter y ghpDetail, y TsgcGridHitInfo añade Line (la línea de visualización, -1 fuera de las líneas), Group (el grupo de la línea, el chip en ghpGroupChip, la posición de destino en ghpGroupPanel), DetailRow y DetailCol.

Eventos

Evento	Tipo	Descripción
OnGetCellStyle	TsgcGridCellStyleEvent	Se lanza el último para cada celda dibujada, con el bloqueo de los datos retenido; Sender es el renderizador.
OnGetGroupText	TsgcGridGetGroupTextEvent	Cambia el título de una fila de grupo. Se genera con el bloqueo de datos retenido.

Los procedimientos sgcGridRender, sgcGridSaveToPNG y sgcGridSaveToSVG hacen lo mismo con un renderizador temporal y un TsgcGridRenderOptions.
sgcGridColumnCaption devuelve el título de una columna en el panel de grupos y en las filas de grupo: Caption y, si no, FieldName.

Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcGrid_Columns, sgcGrid_Data, sgcGrid_Format,
  sgcGrid_Render;

var
  oData: TsgcGridData;
  oOptions: TsgcGridRenderOptions;
  vState: TsgcGridRenderState;
```

```
begin
  // nil: the data creates and owns its own columns
  oData := TsgcGridData.Create(nil);
  oOptions := TsgcGridRenderOptions.Create;
  try
    // '.' decimals and ',' thousands whatever the locale
    oData.UseInvariantFormat;
    oData.Columns.Add('Symbol', 'Symbol');
    oData.Columns.Add('Price', 'Price', gdtFloat).DisplayFormat := '#,##0.00';
    oData.Columns.Add('Change', 'Change %', gdtFloat).DisplayFormat :=
      '+0.00;-0.00;0.00';
    oData.Columns.Add('Volume', 'Volume', gdtInteger).DisplayFormat := '#,##0';
    oData.Columns.Add('Halted', 'Halted', gdtBoolean).Width := 70;
    oData.AppendRow('MSFT', ['MSFT', 412.5, 1.25, 1200500, False]);
    oData.AppendRow('AAPL', ['AAPL', 228.1, -0.84, 3400200, False]);
    oData.AppendRow('GOOG', ['GOOG', 171.9, 0.37, 980100, False]);
    oData.AppendRow('AMZN', ['AMZN', 186.4, -2.10, 2250000, True]);
    oData.AppendRow('NVDA', ['NVDA', 121.7, 3.02, 5120750, False]);
    oData.Columns[1].SortOrder := gsoDescending;

    oOptions.Theme := utDark;
    oOptions.FormatRules.Add('Change', gfoLess, '0', 0,
      sgcUIARGB($FF, $F0, $60, $60));
    oOptions.FormatRules.Add('Volume', gfoGreater, '5000000', 0, 0, True);

    vState := sgcGridRenderState;
    vState.SelectedRow := 1;
    // 470 x 200 logical pixels, written at scale 2 as 940 x 400 pixels
    sgcGridSaveToPNG(oData, oOptions, vState, 'grid.png', 470, 200, 2);
    sgcGridSaveToSVG(oData, oOptions, vState, 'grid.svg', 470, 200);
  finally
    oOptions.Free;
    oData.Free;
  end;
end;
```

Consulte también

- Renderizado sin interfaz
- Temas
- Formato condicional
- Agrupación
- Columnas fijas
- TsgcGridLineView

TsgcGridFormatRule

Una regla de formato condicional, un elemento de TsgcGridFormatRules, en la unidad sgcGrid_Format.

Propiedades

Propiedad	Tipo	Descripción
FieldName	string	La columna cuyo valor se evalúa.
Operator	TsgcGridFormatOperator	gfoEqual, gfoNotEqual, gfoGreater, gfoGreaterEqual, gfoLess, gfoLessEqual, gfoBetween, gfoContains, gfoStartsWith o gfoIsEmpty.
Value, Value2	string	El valor con el que comparar y el límite superior de gfoBetween (inclusive).
BackColor, ForeColor	TsgcUIColor	Los colores de una celda coincidente; 0 los deja sin cambios.
Bold	Boolean	Texto en negrita para una celda coincidente.
ApplyToRow	Boolean	Aplica estilo a todas las celdas de la fila cuando coincide el valor de FieldName.
Enabled	Boolean	True de forma predeterminada.
StopOnMatch	Boolean	Una coincidencia termina la evaluación de esa celda.

Métodos

Método	Devuelve	Descripción
Matches(aValue, aDataType)	Boolean	Evalúa un valor.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupRules(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
var
  oRule: TsgcGridFormatRule;
begin
  // red text for a negative change
  Grid.FormatRules.Add('Change', gfoLess, '0', 0,
    sgcUIARGB($FF, $D0, $30, $30));
  // the whole row tinted when a symbol is halted
  oRule := Grid.FormatRules.Add('Halted', gfoEqual, 'true',
    sgcUIARGB($40, $F0, $A0, $20));
  oRule.ApplyToRow := True;
  // bold prices between 100 and 200, and no later rule for that cell
  oRule := Grid.FormatRules.Add;
```

```
  oRule.FieldName := 'Price';
  oRule.Operator := gfoBetween;
  oRule.Value := '100';
  oRule.Value2 := '200';
  oRule.Bold := True;
  oRule.StopOnMatch := True;
  // the last word, per painted cell
  Grid.OnGetCellStyle := Handlers.GetCellStyle;
end;
```

Consulte también

- Formato condicional
- TsgcGridFormatRules

TsgcGridFormatRules

La colección de elementos TsgcGridFormatRule, publicada como FormatRules por las cuadrículas y por TsgcGridRenderOptions.

Métodos

Método	Devuelve	Descripción
Add	TsgcGridFormatRule	Añade una regla vacía.
Add(aFieldName, aOperator, aValue, aBackColor, aForeColor, aBold)	TsgcGridFormatRule	Añade una regla en una sola llamada; aForeColor y aBold son opcionales.
Prepare, EvaluateRow, ApplyToCell		Lo usa el renderizador mientras dibuja.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Cambió una regla.

Consulte también

- Formato condicional
- TsgcGridFormatRule

IsgcGridDataHost e IsgcUITable

IsgcGridDataHost

Una interfaz de la unidad sgcGrid_Data con un solo método, GetGridData, implementada por todo componente que posee un TsgcGridData: TsgcVCLGrid y TsgcFMXGrid. Los orígenes de datos y la consulta de IA obtienen los datos de su propiedad **Grid** con sgcGridResolveData: primero mediante IsgcGridDataHost, que funciona en todas las versiones de Delphi, incluida Delphi 7, después mediante una propiedad publicada Data y, desde Delphi 2010, mediante una propiedad pública Data. Su propiedad GridData, cuando está asignada, prevalece siempre sobre Grid.

IsgcUITable

La interfaz de tabla de sgcUICore (unidad sgcUICore_Table), implementada por TsgcGridData y delegada en él por las cuadrículas. Permite que otro producto lea la cuadrícula sin depender de sgcGrid, como hace TsgcChartDataSource_Table de sgcCharts. Las filas son filas de vista y las columnas son índices de Columns.

Método	Descripción
TableRowCount, TableColCount	El número de filas de vista y de columnas.
TableColName(aCol)	El FieldName de una columna.
TableValue(aRow, aCol), TableText(aRow, aCol)	Cells y CellText.
TableLock, TableUnlock	Lock y Unlock.
TableVersion	Se incrementa con cada cambio.
TableAddListener, TableRemoveListener	Eventos de notificación lanzados después de OnChange, en el hilo del cambio, con los datos como Sender.

El recuento de referencias está desactivado. Los datos pertenecen a su cuadrícula, por lo que debe soltar la referencia a la interfaz antes de liberar la cuadrícula.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure ReadAsTable(Grid: TsgcVCLGrid);
var
  oTable: IsgcUITable;
  i: Integer;
begin
  // no reference counting: the grid owns the data
  oTable := Grid.Data;
  oTable.TableLock;
  try
    for i := 0 to oTable.TableRowCount - 1 do
      OutputDebugString(PChar(oTable.TableText(i, 0)));
  finally
```

```
    oTable.TableUnlock;
  end;
end;
```

Consulte también

- Gráficos desde la cuadrícula
- TsgcGridData

TsgcGridSource_DataSet

Rellena una cuadrícula desde cualquier TDataSet, en la unidad sgcGrid_Source_DataSet (solo la unidad DB, sin dependencia de VCL ni de FMX).

Propiedades

Propiedad	Tipo	Descripción
Grid	TComponent	La cuadrícula que se rellena. Consulte IsgcGridDataHost.
GridData	TsgcGridData	Unos datos que rellenar en lugar de una cuadrícula. Pública; prevalece sobre Grid.
DataSet	TDataSet	El conjunto de datos de origen.
KeyField	string	El campo cuyo texto es la clave de fila. Con clave, las filas se actualizan en el lugar y conservan su destello, las claves nuevas se añaden y las que faltan se eliminan. Sin clave, cada carga sustituye las filas.
AutoColumns	Boolean	Crea las columnas a partir de los campos cuando la cuadrícula no tiene ninguna. True de forma predeterminada.
LiveUpdates	Boolean	Sigue al conjunto de datos: un Post, una inserción, una eliminación, una cancelación o una actualización vuelve a sincronizar la cuadrícula. Se ignoran las ediciones no confirmadas y el desplazamiento. True de forma predeterminada.
Active	Boolean	Carga el conjunto de datos cuando está abierto y sigue sus cambios.

Métodos

Método	Devuelve	Descripción
Load		Carga el conjunto de datos ahora.
Refresh		Recarga cuando Active.
FieldTypeToDataType(aFieldType)	TsgcGridDataType	El tipo de columna de un tipo de campo. Función de clase.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

function BindDataSet(Grid: TsgcVCLGrid; DataSet: TDataSet)
  : TsgcGridSource_DataSet;
begin
  Result := TsgcGridSource_DataSet.Create(Grid.Owner);
```

```
  Result.Grid := Grid;
  Result.DataSet := DataSet;
  Result.KeyField := 'ID'; // rows are updated in place by ID
  Result.AutoColumns := True; // columns from the fields when the grid has none
  Result.LiveUpdates := True; // follow Post, Delete and Refresh
  Result.Active := True;
end;
```

Consulte también

- Enlace con DataSet
- JSON y CSV

TsgcGridFeedSource

La clase base de los orígenes de flujo de sgcGridWS, en la unidad sgcGridWS_Source. Lee cada mensaje como JSON y escribe sus filas en una cuadrícula por clave: solo se escriben las rutas presentes en una fila, de modo que un mensaje puede llevar una actualización parcial. Los números JSON se escriben como Double, los booleanos como Boolean, las cadenas como string y null como Null.
Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Propiedades

Propiedad	Tipo	Descripción
Grid	TComponent	La cuadrícula en la que se escribe.
GridData	TsgcGridData	Unos datos en los que escribir en lugar de una cuadrícula. Pública; prevalece sobre Grid.
RowsPath	string	La ruta de las filas dentro de un mensaje: una matriz de objetos fila o un solo objeto fila. Vacía: el propio mensaje es una fila o una matriz de filas.
KeyPath	string	La ruta de la clave de fila dentro de un objeto fila, por ejemplo symbol o id.value. Una fila sin clave se omite y se notifica mediante OnError.
Fields	TStrings	Un FieldName=json.path por línea, como Price=quote.last o Bid=book.bid[0]. Una línea sin = lee la ruta del mismo nombre. Vacía: cada columna se lee de la ruta con el nombre de su FieldName.
DeletePath	string	Una ruta booleana en el objeto fila: true, un número distinto de cero o la cadena true elimina la fila de la clave.
MaxRowsPerMessage	Integer	El máximo de filas aplicadas desde un mensaje (10000); 0 sin límite. Las filas sobrantes se descartan y se notifican.
Active	Boolean	False ignora todos los mensajes. True de forma predeterminada.

Eventos

Evento	Tipo	Descripción
OnRow	TsgcGridFeedRowEvent	Acepta u omite la fila de una clave.
OnError	TsgcGridFeedErrorEvent	No se pudo aplicar un mensaje o una fila.

Subprocesos

Todas las filas de un mensaje se aplican dentro de un único Data.BeginUpdate y EndUpdate, en el hilo en el que llega el mensaje. OnRow y OnError también se ejecutan en ese hilo, por lo que debe sincronizar antes de tocar un control desde ellos.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

procedure TFeedHandlers.FeedError(Sender: TObject; const aError: string);
begin
  // runs in the thread the message arrived in
  OutputDebugString(PChar('feed: ' + aError));
end;

procedure TFeedHandlers.FeedRow(Sender: TObject; const aKey: string;
  var aAccept: Boolean);
begin
  // skip the rows of test symbols
  aAccept := Pos('TEST', aKey) <> 1;
end;
```

Consulte también

- TsgcGridFeedSource_WebSocket
- TsgcGridFeedSource_MQTT
- TsgcGridFeedSource_SSE
- Flujos en vivo

TsgcGridFeedSource_WebSocket

Un TsgcGridFeedSource que lee los mensajes de un TsgcWebSocketClient de sgcWebSockets, en la unidad sgcGridWS_Source_WebSocket.
Mientras el origen está Active y Client está asignado, en tiempo de ejecución, instala su propio controlador OnMessage en el cliente y conserva el controlador que había antes: cada mensaje de texto se aplica primero a la cuadrícula y después se pasa al controlador anterior, de modo que el controlador de la aplicación sigue ejecutándose. Al establecer Active en False, quitar Client o destruir el origen se restaura el controlador anterior.
Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Propiedades

Propiedad	Tipo	Descripción
Client	TsgcWebSocketClient	El cliente WebSocket.
Grid	TComponent	La cuadrícula en la que se escribe.
GridData	TsgcGridData	Unos datos en los que escribir en lugar de una cuadrícula. Pública; prevalece sobre Grid.
RowsPath	string	La ruta de las filas dentro de un mensaje: una matriz de objetos fila o un solo objeto fila. Vacía: el propio mensaje es una fila o una matriz de filas.
KeyPath	string	La ruta de la clave de fila dentro de un objeto fila, por ejemplo symbol o id.value. Una fila sin clave se omite y se notifica mediante OnError.
Fields	TStrings	Un FieldName=json.path por línea, como Price=quote.last o Bid=book.bid[0]. Una línea sin = lee la ruta del mismo nombre. Vacía: cada columna se lee de la ruta con el nombre de su FieldName.
DeletePath	string	Una ruta booleana en el objeto fila: true, un número distinto de cero o la cadena true elimina la fila de la clave.
MaxRowsPerMessage	Integer	El máximo de filas aplicadas desde un mensaje (10000); 0 sin límite. Las filas sobrantes se descartan y se notifican.
Active	Boolean	False ignora todos los mensajes. True de forma predeterminada.

Métodos

Método	Devuelve	Descripción
ProcessMessage(aText)	Integer	Aplica un mensaje manualmente y devuelve el número de filas aplicadas.

Eventos

Evento	Tipo	Descripción

OnRow	TsgcGridFeedRowEvent	Acepta u omite la fila de una clave.
OnError	TsgcGridFeedErrorEvent	No se pudo aplicar un mensaje o una fila.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function FeedFromWebSocket(Grid: TsgcVCLGrid; Client: TsgcWebSocketClient)
  : TsgcGridFeedSource_WebSocket;
begin
  // messages such as {"quotes":[{"symbol":"MSFT","quote":{"last":412.5}}]}
  Result := TsgcGridFeedSource_WebSocket.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.RowsPath := 'quotes';
  Result.KeyPath := 'symbol';
  Result.Fields.Add('Symbol=symbol');
  Result.Fields.Add('Price=quote.last');
  Result.Fields.Add('Volume'); // same-named path
  Result.DeletePath := 'removed';
  Result.Client := Client;
  Client.URL := 'wss://quotes.example.com/stream';
  Client.Active := True;
end;
```

Consulte también

- TsgcGridFeedSource
- Flujos en vivo

TsgcGridFeedSource_MQTT

Un TsgcGridFeedSource que lee los mensajes de un TsgcWSPClient_MQTT de sgcWebSockets, en la unidad sgcGridWS_Source_MQTT.
Mientras el origen está Active y Client está asignado, en tiempo de ejecución, instala su propio controlador OnMQTTPublish en el cliente y conserva el controlador que había antes: cada publicación cuyo tema coincide con Topic se aplica primero a la cuadrícula y después se pasa al controlador anterior, de modo que el controlador de la aplicación sigue ejecutándose. Al establecer Active en False, quitar Client o destruir el origen se restaura el controlador anterior. El origen no se suscribe: la aplicación se suscribe en el cliente.
Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Propiedades

Propiedad	Tipo	Descripción
Client	TsgcWSPClient_MQTT	El cliente MQTT.
Topic	string	Un filtro de tema MQTT: + coincide con un nivel, # con este nivel y todos los inferiores. Vacío acepta todas las publicaciones.
Grid	TComponent	La cuadrícula en la que se escribe.
GridData	TsgcGridData	Unos datos en los que escribir en lugar de una cuadrícula. Pública; prevalece sobre Grid.
RowsPath	string	La ruta de las filas dentro de un mensaje: una matriz de objetos fila o un solo objeto fila. Vacía: el propio mensaje es una fila o una matriz de filas.
KeyPath	string	La ruta de la clave de fila dentro de un objeto fila, por ejemplo symbol o id.value. Una fila sin clave se omite y se notifica mediante OnError.
Fields	TStrings	Un FieldName=json.path por línea, como Price=quote.last o Bid=book.bid[0]. Una línea sin = lee la ruta del mismo nombre. Vacía: cada columna se lee de la ruta con el nombre de su FieldName.
DeletePath	string	Una ruta booleana en el objeto fila: true, un número distinto de cero o la cadena true elimina la fila de la clave.
MaxRowsPerMessage	Integer	El máximo de filas aplicadas desde un mensaje (10000); 0 sin límite. Las filas sobrantes se descartan y se notifican.
Active	Boolean	False ignora todos los mensajes. True de forma predeterminada.

Métodos

Método	Devuelve	Descripción
ProcessPublish(aTopic, aText)	Integer	Aplica una publicación manualmente.

Eventos

Evento	Tipo	Descripción
OnRow	TsgcGridFeedRowEvent	Acepta u omite la fila de una clave.
OnError	TsgcGridFeedErrorEvent	No se pudo aplicar un mensaje o una fila.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function FeedFromMQTT(Grid: TsgcVCLGrid; MQTT: TsgcWSPClient_MQTT)
  : TsgcGridFeedSource_MQTT;
begin
  Result := TsgcGridFeedSource_MQTT.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.KeyPath := 'id';
  Result.Topic := 'plant/+/sensors';
  Result.Client := MQTT;
  // the source does not subscribe: subscribe on the MQTT client, for
  // example in its OnMQTTConnect event
end;
```

Consulte también

- TsgcGridFeedSource
- Flujos en vivo

TsgcGridFeedSource_SSE

Un TsgcGridFeedSource que lee los mensajes de los eventos enviados por el servidor de un TsgcHTTP1Client de sgcWebSockets, en la unidad sgcGridWS_Source_SSE.
Mientras el origen está Active y Client está asignado, en tiempo de ejecución, instala su propio controlador OnSSEMessage en el cliente y conserva el controlador que había antes: cada fragmento del flujo se aplica primero a la cuadrícula y después se pasa al controlador anterior, de modo que el controlador de la aplicación sigue ejecutándose. Al establecer Active en False, quitar Client o destruir el origen se restaura el controlador anterior. Los fragmentos se dividen en eventos: las líneas de datos se unen, una línea vacía termina el evento y un evento puede abarcar varios fragmentos. La aplicación inicia el flujo con Client.GetSSE.
Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Propiedades

Propiedad	Tipo	Descripción
Client	TsgcHTTP1Client	El cliente HTTP.
Grid	TComponent	La cuadrícula en la que se escribe.
GridData	TsgcGridData	Unos datos en los que escribir en lugar de una cuadrícula. Pública; prevalece sobre Grid.
RowsPath	string	La ruta de las filas dentro de un mensaje: una matriz de objetos fila o un solo objeto fila. Vacía: el propio mensaje es una fila o una matriz de filas.
KeyPath	string	La ruta de la clave de fila dentro de un objeto fila, por ejemplo symbol o id.value. Una fila sin clave se omite y se notifica mediante OnError.
Fields	TStrings	Un FieldName=json.path por línea, como Price=quote.last o Bid=book.bid[0]. Una línea sin = lee la ruta del mismo nombre. Vacía: cada columna se lee de la ruta con el nombre de su FieldName.
DeletePath	string	Una ruta booleana en el objeto fila: true, un número distinto de cero o la cadena true elimina la fila de la clave.
MaxRowsPerMessage	Integer	El máximo de filas aplicadas desde un mensaje (10000); 0 sin límite. Las filas sobrantes se descartan y se notifican.
Active	Boolean	False ignora todos los mensajes. True de forma predeterminada.

Métodos

Método	Devuelve	Descripción
ProcessChunk(aText)	Integer	Aplica manualmente un fragmento del flujo.
ResetStream		Descarta un evento recibido a medias.

Eventos

Evento	Tipo	Descripción
OnRow	TsgcGridFeedRowEvent	Acepta u omite la fila de una clave.
OnError	TsgcGridFeedErrorEvent	No se pudo aplicar un mensaje o una fila.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function FeedFromSSE(Grid: TsgcVCLGrid; HTTP: TsgcHTTP1Client)
  : TsgcGridFeedSource_SSE;
begin
  Result := TsgcGridFeedSource_SSE.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.KeyPath := 'orderId';
  Result.Client := HTTP;
  // the application starts the stream
  HTTP.GetSSE('https://api.example.com/orders/stream');
end;
```

Consulte también

- TsgcGridFeedSource
- Flujos en vivo

TsgcGridAIQuery

Convierte una pregunta en lenguaje natural en filtros, ordenaciones y resaltados de una cuadrícula, en la unidad sgcGridWS_AI. La pregunta la resuelve el TsgcHTMLAIAssistant de sgcWebSockets: un LLM cuando el Assistant tiene un proveedor, o si no su analizador basado en reglas, que entiende preguntas como price greater than 100, sort by volume descending y highlight change less than 0. La respuesta se comprueba con las columnas de la cuadrícula antes de aplicar nada.
Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Propiedades

Propiedad	Tipo	Descripción
Grid	TComponent	La cuadrícula que se consulta. Consulte IsgcGridDataHost.
GridData	TsgcGridData	Unos datos que consultar en lugar de una cuadrícula. Pública; prevalece sobre Grid.
Assistant	TsgcHTMLAIAssistant	El asistente que responde. Nil usa el analizador basado en reglas, sin red y sin clave de API.
Active	Boolean	True de forma predeterminada.

Métodos

Método	Devuelve	Descripción
Ask(aQuestion)	Boolean	Elimina lo que aplicó el Ask anterior y después aplica la respuesta. Síncrono; llámelo desde el hilo principal.
Reset		Restaura el OnFilterRow y la ordenación anteriores, y elimina solo las reglas de formato que añadió este componente.
LastExplanation	string	Qué se aplicó y qué se ignoró.

Eventos

Evento	Tipo	Descripción
OnApplied	TNotifyEvent	Se aplicó una respuesta.
OnError	TsgcGridAIErrorEvent	No se pudo responder o aplicar la pregunta.

Ejemplo

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
```

```
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function AskGrid(Grid: TsgcVCLGrid): string;
var
  oQuery: TsgcGridAIQuery;
begin
  oQuery := TsgcGridAIQuery.Create(Grid.Owner);
  oQuery.Grid := Grid;
  // no Assistant: the rule-based parser answers, no network, no API key
  if oQuery.Ask('price greater than 100 sort by volume descending ' +
    'highlight change less than 0') then
    Result := oQuery.LastExplanation
  else
    Result := 'not understood';
  // later: remove the filter, the sort and the highlight
  oQuery.Reset;
end;
```

Consulte también

- Pregunte a su cuadrícula
- Filtrado
- Formato condicional

Actualizaciones en vivo y subprocesos

sgcGrid está pensado para filas que cambian todo el tiempo. Cada fila tiene una clave, y un escritor actualiza la fila de una clave por nombre de campo, desde cualquier hilo, sin Synchronize.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

Actualizar por clave

UpdateRow establece los campos indicados de la fila de una clave e inserta la fila cuando la clave es nueva (salvo que AutoInsert sea False). UpdateCell establece un único campo, AppendRow añade una fila con los valores en el orden de las columnas y DeleteRow elimina la fila de una clave. Cada celda modificada registra la hora de su cambio y su dirección, arriba, abajo o modificada, que gobierna el destello al cambiar.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

type
  TFeedThread = class(TThread)
  private
    FData: TsgcGridData;
  protected
    procedure Execute; override;
  public
    constructor Create(aData: TsgcGridData);
  end;

constructor TFeedThread.Create(aData: TsgcGridData);
begin
  FData := aData;
  inherited Create(False);
end;

procedure TFeedThread.Execute;
const
  CS_SYMBOLS: array [0 .. 2] of string = ('MSFT', 'AAPL', 'GOOG');
begin
  while not Terminated do
  begin
    // thread-safe: no Synchronize, the grid repaints at most MaxFPS
    // times per second in the main thread
    FData.UpdateRow(CS_SYMBOLS[Random(3)], ['Price', 'Time'],
      [100 + Random * 50, Now]);
    Sleep(1);
  end;
end;
```

Repintados

Los datos lanzan OnChange en el hilo escritor. La cuadrícula solo se marca como sucia en él y se repinta en el hilo principal como máximo MaxFPS veces por segundo, de modo que 1.000 actualizaciones por segundo cuestan como máximo 60 fotogramas. La fila seleccionada se vuelve a localizar por su clave tras cada cambio de vista.

BeginUpdate y EndUpdate

Envuelva un lote en BeginUpdate y EndUpdate para obtener una única notificación de todo él.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure LoadSnapshot(Grid: TsgcVCLGrid);
var
  i: Integer;
begin
  Grid.Data.BeginUpdate;
  try
    Grid.Data.Clear;
    for i := 1 to 1000 do
      Grid.Data.AppendRow('S' + IntToStr(i), ['S' + IntToStr(i), 100.0, 0.0,
        0, Now, False]);
  finally
    // one change notification and one repaint for the whole batch
    Grid.Data.EndUpdate;
  end;
end;
```

Modo cola

Para registros y flujos que crecen, **MaxRows** mayor que 0 elimina las filas más antiguas cuando se insertan otras nuevas, y **TailMode** mantiene la fila más reciente a la vista mientras la cuadrícula está desplazada hasta el final. Desplazarse hacia arriba deja de seguirla y volver al final la reanuda.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupLog(Grid: TsgcVCLGrid);
begin
  Grid.Columns.Clear;
  Grid.Columns.Add('Level', 'Level').Width := 60;
  Grid.Columns.Add('Text', 'Message').Width := 400;
  Grid.Columns.Add('Time', 'Time', gdtDateTime).DisplayFormat := 'hh:nn:ss.zzz';
  Grid.MaxRows := 5000; // keep the newest 5,000 rows
  Grid.TailMode := True; // follow the newest row while at the bottom
  // an empty key generates a unique key '#<n>'
  Grid.Data.AppendRow('', ['info', 'Service started', Now]);
end;
```

Consulte también

- TsgcGridData
- Rendimiento
- Flujos en vivo

Ordenación

Cada columna tiene un SortOrder (gsoNone, gsoAscending o gsoDescending) y un SortIndex, su posición en una ordenación por varias columnas. La ordenación es estable. El texto se compara sin distinguir mayúsculas de minúsculas y los demás tipos numéricamente; los valores nulos, vacíos o no convertibles se ordenan siempre primero.

Con el ratón

Un clic en un encabezado recorre la columna entre ascendente, descendente y sin ordenar. Con Ctrl pulsado, la columna se añade a la ordenación actual en lugar de reemplazarla, y el encabezado muestra la posición de cada columna ordenada. OnSortChange se lanza después del cambio.

Desde código

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SortByChange(Grid: TsgcVCLGrid);
begin
  Grid.Data.BeginUpdate;
  try
    // first by Change, highest first, then by Symbol
    Grid.Columns.FindByField('Change').SortIndex := 0;
    Grid.Columns.FindByField('Change').SortOrder := gsoDescending;
    Grid.Columns.FindByField('Symbol').SortIndex := 1;
    Grid.Columns.FindByField('Symbol').SortOrder := gsoAscending;
  finally
    Grid.Data.EndUpdate;
  end;
end;
```

Regla en vivo

Un cambio de valor reordena la vista solo cuando su columna es una columna de ordenación o hay un filtro activo. Una columna que cambia constantemente pero no está ordenada conserva el orden actual y no cuesta ninguna reordenación. Las inserciones, las eliminaciones y los cambios de ordenación o de filtro siempre reconstruyen la vista.

Consulte también

- Filtrado
- TsgcGridColumn

Filtrado

Se aplican dos filtros, uno tras otro. **FilterText** conserva las filas cuyo texto formateado, en cualquier columna visible, contiene el texto, sin distinguir mayúsculas de minúsculas. Después se consulta **OnFilterRow** de los datos por cada fila que ha pasado.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure FilterGrid(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
begin
  // rows whose visible text contains 'ms', case-insensitive
  Grid.FilterText := 'ms';
  // and a predicate on top of it
  Grid.Data.OnFilterRow := Handlers.FilterRow;
  // after the logic of the handler changed, rebuild the view
  Grid.Data.InvalidateView;
end;
```

OnFilterRow

El controlador recibe una fila del almacén, no una fila de vista: lea sus valores con StoreCells o su clave con StoreKeyOf. Se llama con el bloqueo de los datos retenido, por lo que no debe esperar a otro hilo que use la cuadrícula. Cuando cambie la lógica del controlador, llame a InvalidateView.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure TGridHandlers.FilterRow(Sender: TObject; aRow: Integer;
  var aAccept: Boolean);
var
  vPrice: Variant;
begin
  // Sender is the TsgcGridData and aRow a store row: read it with
  // StoreCells, the lock is already held
  vPrice := TsgcGridData(Sender).StoreCells[aRow, 1];
  aAccept := not VarIsNull(vPrice) and (vPrice > 100);
end;
```

Con un filtro activo, el cambio de cualquier valor puede mover una fila dentro o fuera de la vista. RowOfKey devuelve -1 para una fila filtrada.

Consulte también

- Ordenación
- Pregunte a su cuadrícula
- TsgcGridData

Agrupación

La agrupación ordena las filas de una cuadrícula bajo filas de grupo, un nivel por cada columna agrupada. Una fila de grupo muestra el miembro de su nivel, el número de filas que contiene y, con los resúmenes, sus totales. Las filas de grupo se expanden y se contraen, y los niveles anidados se sangran GroupIndent DIP (20 por defecto). La agrupación forma parte de todas las ediciones, incluida Community, y funciona igual en TsgcVCLGrid y TsgcFMXGrid.
La agrupación es estado de la vista. La cuadrícula guarda sus filas de grupo en un TsgcGridLineView (Lines), mientras que TsgcGridData, sus filas de la vista e IsgcUITable no cambian, de modo que un gráfico o cualquier otro lector enlazado a la cuadrícula sigue viendo las filas planas, ordenadas y filtradas. Sin ninguna columna agrupada, la cuadrícula se comporta exactamente como una cuadrícula sin agrupar.

Agrupar una columna

Cada columna tiene un GroupIndex, -1 por defecto. Un valor de 0 o más agrupa la cuadrícula por esa columna en ese nivel: los niveles se ordenan por GroupIndex y después por el Index de la columna. Una columna oculta también agrupa. GroupBy asigna GroupIndex a las columnas indicadas (FieldName, sin distinguir mayúsculas, nivel 0 primero) y -1 a todas las demás; los campos desconocidos se ignoran. ClearGrouping asigna GroupIndex -1 a todas las columnas. Cambiar GroupIndex no reconstruye la vista de los datos.

- Los miembros de un nivel se ordenan por el valor tipado de su primera fila, con la regla de la ordenación de la cuadrícula: el texto sin distinguir mayúsculas, los demás tipos numéricamente, en orden ascendente, o descendente cuando el SortOrder de la columna es gsoDescending.
- Null, el texto vacío y los valores que no se convierten forman el miembro vacío, que se muestra como (blank) y va siempre primero.
- Cuando la columna tiene un DisplayFormat, el miembro es el texto formateado de la celda, de modo que un DisplayFormat de yyyy-mm agrupa las fechas por mes.
- Dentro de un grupo las filas conservan el orden de la ordenación de la cuadrícula y el filtro sigue aplicándose.

El panel de grupos

ShowGroupPanel muestra un panel de GroupPanelHeight DIP (36 por defecto) sobre el encabezado, con un chip por cada nivel agrupado, o una indicación cuando no hay nada agrupado. Con el ratón:

- Arrastre el encabezado de una columna más de 4 píxeles sobre el panel para agrupar por ella en la posición donde lo suelte. Los encabezados solo se arrastran mientras el panel está visible, y Escape cancela el arrastre.
- Arrastre un chip dentro del panel para mover su nivel, o fuera del panel para desagrupar su columna.
- Haga clic en un chip para alternar el orden de sus grupos. Esto invierte el SortOrder de la columna y genera OnSortChange.

En TsgcFMXGrid un gesto táctil de desplazamiento sigue desplazando la cuadrícula y cancela el arrastre de un encabezado, de modo que agrupar con el tacto se hace por código o con el ratón.

Filas de grupo

Una fila de grupo muestra un chevron, el título y los resúmenes de grupo de las columnas con un SummaryType (GroupSummaries, True por defecto). El título es el título de la columna, el miembro y el número de filas, como en Region: North (12). OnGetGroupText puede sustituirlo; se genera con el bloqueo de datos retenido, así que lea la cuadrícula allí pero no escriba nunca en los datos. GroupInfo(aGroup) devuelve el nivel, el grupo padre, la columna agrupada, el texto y el valor del miembro, el número de filas, el estado expandido y la línea de visualización de un grupo, vea TsgcGridLineView.

Expandir y contraer

GroupsExpanded (True por defecto) es el estado de los grupos nuevos; al asignarlo se conserva el estado de los grupos actuales. ExpandAll y CollapseAll cambian todos los grupos y los grupos que se creen después. ExpandGroup y CollapseGroup cambian un solo grupo. OnGroupExpandChange indica el grupo y su nuevo estado, con

aGroup -1 tras ExpandAll y CollapseAll. El estado expandido de un grupo se conserva entre actualizaciones en vivo y reordenaciones.
Un clic en el chevron o un doble clic en una fila de grupo la alterna. Teclas en una línea de grupo:

Tecla	Acción
Right	Expande el grupo, o pasa a su primera línea hija cuando ya está expandido.
Left	Contrae el grupo, o pasa a su grupo padre cuando ya está contraído.
+ y -	Expanden y contraen el grupo.
*	Expande el grupo y todos los grupos que contiene.
Enter	Alterna el grupo.
Ctrl+C	Copia el título de la fila de grupo.

Cuando una contracción oculta la fila seleccionada, su línea de grupo visible recibe el foco.

Líneas y selección

Mientras las filas están agrupadas, la cuadrícula se desplaza por líneas de visualización: TopRow y la barra de desplazamiento vertical cuentan líneas, y FocusedLine es la línea con el foco, -1 si no hay ninguna. Los índices de fila siguen siendo filas de la vista en todos los demás sitios (SelectedRow, Cells, OnCellClick). En una línea de grupo SelectedRow es -1. La fila seleccionada se conserva por su clave y se vuelve a encontrar tras reagrupar. GroupCount devuelve el número de grupos y Lines da la vista de líneas completa.
Un cambio de valor en vivo en una columna agrupada mueve la fila a su nuevo grupo, y OnGroupingChange se genera siempre que cambian las columnas agrupadas o sus niveles, desde el panel o desde código.

Ejemplo

Los controladores viven en una clase pequeña, como los controladores de los demás ejemplos de características.

```
uses
  SysUtils, Classes, Forms, Controls,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_Lines, sgcGrid_VCL;

type
  TGroupHandlers = class
  public
    procedure GetGroupText(Sender: TObject; aGroup: Integer;
      var aText: string);
    procedure GroupExpandChange(Sender: TObject; aGroup: Integer;
      aExpanded: Boolean);
    procedure GroupingChange(Sender: TObject);
  end;

procedure TGroupHandlers.GetGroupText(Sender: TObject; aGroup: Integer;
  var aText: string);
var
  vInfo: TsgcGridGroupInfo;
begin
  // raised with the data lock held: read, never write the data here
  vInfo := TsgcVCLGrid(Sender).GroupInfo(aGroup);
  aText := Format('%s (%d orders)', [vInfo.Text, vInfo.RowCount]);
end;

procedure TGroupHandlers.GroupExpandChange(Sender: TObject; aGroup: Integer;
  aExpanded: Boolean);
begin
  // aGroup is -1 after ExpandAll and CollapseAll
```

```
end;

procedure TGroupHandlers.GroupingChange(Sender: TObject);
begin
  // the grouped columns or their order changed
end;

procedure GroupOrders(Grid: TsgcVCLGrid; Handlers: TGroupHandlers);
var
  i: Integer;
begin
  Grid.OnGetGroupText := Handlers.GetGroupText;
  Grid.OnGroupExpandChange := Handlers.GroupExpandChange;
  Grid.OnGroupingChange := Handlers.GroupingChange;
  Grid.ShowGroupPanel := True;
  // new groups start collapsed
  Grid.GroupsExpanded := False;
  // level 0 by Region, level 1 by Product
  Grid.GroupBy(['Region', 'Product']);
  // open the outer level only
  for i := 0 to Grid.GroupCount - 1 do
    if Grid.GroupInfo(i).Level = 0 then
      Grid.ExpandGroup(i);
end;

procedure GroupByColumn(Grid: TsgcVCLGrid);
begin
  // the same through the columns: GroupIndex is the level, -1 ungroups
  Grid.Columns.FindByField('Region').GroupIndex := 0;
  Grid.Columns.FindByField('Product').GroupIndex := 1;
end;
```

Consulte también

- Resúmenes
- Columnas fijas
- Ordenación
- TsgcGridLineView
- TsgcGridColumn
- TsgcVCLGrid
- Agrupación y resúmenes de .NET

Resúmenes

Una columna con un SummaryType muestra un resumen de sus valores en la fila de pie y en cada fila de grupo. Los resúmenes forman parte de todas las ediciones, incluida Community.

SummaryType	Valor
gsuNone	Sin resumen (por defecto).
gsuSum	La suma de los valores numéricos.
gsuCount	El número de valores que no son Null ni texto vacío.
gsuAverage	La suma dividida por el número de valores numéricos.
gsuMin, gsuMax	El menor y el mayor valor.

- Null, Unassigned y el texto vacío se omiten. El texto se convierte con el FormatSettings de los datos, igual que lo convierte el modelo de datos.
- Las columnas de texto y booleanas solo admiten gsuCount.
- Las columnas de fecha conservan Min y Max como fechas y no tienen Sum ni Average.
- Un resumen sin ningún valor numérico es Null, y entonces gsuCount es 0.

Formato

SummaryFormat es una máscara de FormatFloat o FormatDateTime. El texto literal va entre comillas dobles, por ejemplo `"Total: "#,##0.00`. Cuando está vacío se usa el DisplayFormat de la columna y, si no, el valor por defecto del tipo de datos; gsuCount usa `0`. Cambiar SummaryType o SummaryFormat actualiza los resúmenes sin reconstruir la vista de los datos.

Filas de pie y de grupo

ShowFooter (False por defecto) muestra una fila de pie de una altura RowHeight anclada en la parte inferior de la cuadrícula, con el resumen de cada columna que tiene un SummaryType sobre las filas de la vista, de modo que un filtro se aplica. El pie sigue el desplazamiento horizontal y las columnas fijas.
GroupSummaries (True por defecto) muestra los resúmenes de cada grupo, sobre las filas del grupo, alineados a la derecha en su fila de grupo. La columna visible situada más a la izquierda contiene siempre el título: un resumen en ella se dibuja después del título si cabe y, si no, solo lo muestra el pie. Vea Agrupación.
FooterSummary(aCol) devuelve el valor de pie de una columna (un índice de Columns), Null si no tiene, y FooterSummaryText(aCol) su texto formateado. Los valores de grupo proceden de Lines.GroupSummary(aGroup, aCol). Los resúmenes se calculan una vez y solo se recalculan cuando cambian los datos, de modo que un fotograma que los pinta no cuesta nada extra. En modo virtual los valores se leen mediante OnGetValue para cada fila.

Ejemplo

```
uses
  SysUtils, Classes, Variants, Forms, Controls,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_VCL;

procedure ShowTotals(Grid: TsgcVCLGrid);
var
  vQty, vAmount: TsgcGridColumn;
begin
  vQty := Grid.Columns.FindByField('Qty');
  vQty.SummaryType := gsuSum;
```

```
  vAmount := Grid.Columns.FindByField('Amount');
  vAmount.SummaryType := gsuSum;
  // literal text goes in double quotes
  vAmount.SummaryFormat := '"Total: "#,##0.00';

  Grid.Columns.FindByField('Price').SummaryType := gsuAverage;
  Grid.Columns.FindByField('Order').SummaryType := gsuCount;

  // the footer row and the summaries in the group rows
  Grid.ShowFooter := True;
  Grid.GroupSummaries := True;
end;

function AmountTotal(Grid: TsgcVCLGrid): Double;
var
  vValue: Variant;
begin
  // FooterSummary takes a Columns index and returns Null without a value
  vValue := Grid.FooterSummary(Grid.Columns.FindByField('Amount').Index);
  if VarIsNull(vValue) then
    Result := 0
  else
    Result := vValue;
end;

function AmountTotalText(Grid: TsgcVCLGrid): string;
begin
  // the text of the footer cell, formatted with SummaryFormat
  Result := Grid.FooterSummaryText(Grid.Columns.FindByField('Amount').Index);
end;
```

Sin una cuadrícula, TsgcGridSummaryAccumulator (unidad sgcGrid_Summary) suma valores y devuelve cualquier tipo de resumen con Value, TsgcGridSummaryCache guarda los valores de pie de un TsgcGridData, y sgcGrid-SummaryText formatea un valor con las reglas anteriores. El ejemplo de TsgcGridLineView usa sgcGridSummary-Text.

Consulte también

- Agrupación
- TsgcGridColumn
- TsgcGridLineView
- Agrupación y resúmenes de .NET

Filas de árbol

Las filas de árbol muestran las filas de una cuadrícula como una jerarquía. Cada fila nombra a su padre mediante una columna clave, y la cuadrícula dispone las filas en profundidad bajo nodos de árbol expandibles, sangradas en una columna. El árbol funciona con filas en memoria, filas cargadas desde un TDataSet y filas modificadas por actualizaciones en vivo, tanto en TsgcVCLGrid como en TsgcFMXGrid.
Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.

Activar el árbol

La propiedad Tree de la cuadrícula (TsgcGridTreeOptions, unidad sgcGrid_Tree) contiene los ajustes. El modo árbol está activo cuando KeyField y ParentField nombran columnas de la cuadrícula; borre una de ellas para desactivarlo. Las columnas de clave y de padre pueden estar ocultas.

Propiedad	Valor predeterminado	Descripción
KeyField		La columna que contiene la clave de cada fila.
ParentField		La columna que contiene la clave de la fila padre. Un valor vacío o Null convierte la fila en raíz.
TreeField		La columna que muestra la sangría y los botones del árbol. Vacío usa la primera columna visible.
Indent	16	La sangría de un nivel, en DIP.
Expanded	False	El estado de los nodos que nunca se expandieron ni se contrajeron. Cambiarlo expande o contrae todos los nodos.

Cómo se enlazan las filas

- Las claves se comparan como texto, distinguiendo mayúsculas. Null y el texto vacío son la clave vacía. Los enteros y los números reales sin fracción usan su texto entero, de modo que 5, 5.0 y '5' son la misma clave. Los demás números reales y las fechas usan el texto del FormatSettings de los datos.
- Los padres se buscan entre todas las filas almacenadas. Cuando varias filas comparten una clave, la primera fila almacenada es la propietaria de los hijos.
- Una fila cuyo padre está vacío, es desconocido (huérfana) o es la propia fila se convierte en raíz. Un ciclo se rompe en la primera fila que se repite, que pasa a ser raíz.
- Los hermanos conservan el orden de la ordenación de la cuadrícula, de modo que ordenar una columna ordena los hijos de cada nodo y las raíces.
- Con un filtro, una fila visible cuyo padre queda filtrado cuelga de su antepasado visible más próximo, o se convierte en raíz cuando no hay ninguno.
- Un origen de datos virtual (Data.Virtual) no muestra árbol.

Expandir y contraer

ExpandNode(aRow) y CollapseNode(aRow) cambian el nodo de la fila de la vista aRow; una fila sin hijos se deja como está. En modo árbol, ExpandAll y CollapseAll actúan sobre los nodos del árbol y sobre los nodos que se creen después. OnTreeExpandChange indica la fila de la vista y su nuevo estado, con aRow -1 tras ExpandAll y CollapseAll. El estado se conserva por clave, de modo que sobrevive a una ordenación, un filtro, una recarga de las filas y las actualizaciones en vivo. Cuando una contracción oculta la fila seleccionada, se selecciona su padre visible más próximo.
Un clic en un botón del árbol alterna el nodo. Teclas en una fila seleccionada en modo árbol:

Tecla	Acción
Right	Expande el nodo, o pasa a su primer hijo cuando ya está expandido.
Left	Contrae el nodo, o pasa a su padre cuando ya está contraído.
+ y -	Expanden y contraen el nodo.
*	Expande el nodo y todos los nodos que contiene.
Ctrl+Right y Ctrl+Left	Expanden y contraen la banda de detalle de la fila, vea Maestro detalle.

En TsgcFMXGrid un toque en un botón del árbol lo alterna, y Cmd funciona como Ctrl.

El árbol y otras características

- La agrupación se ignora mientras el árbol está activo. El GroupIndex de las columnas se conserva y vuelve a aplicarse cuando se desactiva el modo árbol.
- Las bandas de detalle funcionan en modo árbol: cada fila maestra expandida muestra su banda debajo.
- SaveToXLSX escribe el árbol como un esquema de Excel, vea Exportación a Excel.
- Los índices de fila siguen siendo filas de la vista (SelectedRow, Cells, OnCellClick). Lines.TreeMode indica si el árbol está activo, y Lines.Tree (TsgcGridTreeBuilder) responde Depth, ParentRow, HasChildren, ChildCount y DescendantCount para una fila de la vista, vea TsgcGridLineView.

Ejemplo

Un organigrama en memoria. Los controladores viven en una clase pequeña, como los controladores de los demás ejemplos de características.

```
uses
  SysUtils, Classes, Variants,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_VCL;

type
  TTreeHandlers = class
  public
    procedure TreeExpandChange(Sender: TObject; aRow: Integer;
      aExpanded: Boolean);
  end;

procedure TTreeHandlers.TreeExpandChange(Sender: TObject; aRow: Integer;
  aExpanded: Boolean);
begin
  // aRow is the view row of the node, -1 after ExpandAll and CollapseAll
end;

procedure ShowOrgChart(Grid: TsgcVCLGrid; Handlers: TTreeHandlers);
begin
  Grid.Columns.Clear;
  // the key and parent columns may be hidden
  Grid.Columns.Add('id', 'Id', gdtInteger).Visible := False;
  Grid.Columns.Add('manager', 'Manager', gdtInteger).Visible := False;
  Grid.Columns.Add('name', 'Name');
  Grid.Columns.Add('title', 'Title');
  Grid.Data.BeginUpdate;
  try
    Grid.Data.Clear;
    // values by column index: id, manager, name, title
    Grid.Data.AppendRow('1', [1, Null, 'Ana Ruiz', 'CEO']);
    Grid.Data.AppendRow('2', [2, 1, 'Marc Puig', 'CTO']);
    Grid.Data.AppendRow('3', [3, 1, 'Laura Vidal', 'CFO']);
    Grid.Data.AppendRow('4', [4, 2, 'Jordi Mas', 'Developer']);
```

```
    Grid.Data.AppendRow('5', [5, 2, 'Eva Soler', 'Developer']);
  finally
    Grid.Data.EndUpdate;
  end;
  Grid.OnTreeExpandChange := Handlers.TreeExpandChange;
  // KeyField and ParentField switch the tree mode on
  Grid.Tree.KeyField := 'id';
  Grid.Tree.ParentField := 'manager';
  // the column that shows the indentation and the tree buttons
  Grid.Tree.TreeField := 'name';
  Grid.Tree.Indent := 20;
  // collapse every node, then open the root
  Grid.CollapseAll;
  Grid.ExpandNode(Grid.Data.RowOfKey('1'));
end;

function NodeDepth(Grid: TsgcVCLGrid; const aKey: string): Integer;
begin
  // 0 for a root, -1 when the grid is not in tree mode
  Result := -1;
  if Grid.Lines.TreeMode then
    Result := Grid.Lines.Tree.Depth(Grid.Data.RowOfKey(aKey));
end;
```

Consulte también

- Maestro detalle
- Agrupación
- Exportación a Excel
- TsgcVCLGrid
- TsgcGridLineView
- Ediciones
- Filas de árbol y maestro detalle de .NET

Maestro detalle

Las filas maestro detalle muestran una banda de detalle bajo cada fila maestra expandida. La banda es una pequeña cuadrícula propia con las filas relacionadas: los pedidos de un cliente, las líneas de una factura. Tiene encabezado, se desplaza y se ordena por sí sola. Las bandas funcionan en modo plano, agrupado y árbol, tanto en TsgcVCLGrid como en TsgcFMXGrid.
Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.

Opciones de detalle

La propiedad Detail de la cuadrícula (TsgcGridDetailOptions, unidad sgcGrid_Detail) contiene los ajustes.

Propiedad	Valor predeterminado	Descripción
Active	False	Activa las bandas de detalle.
Source	nil	El componente que contiene las filas de detalle: un TsgcGridDataStore o una cuadrícula. La cuadrícula lo vigila con una notificación de liberación, de modo que un origen liberado asigna nil a Source.
MasterKeyField		La columna maestra cuyo valor se busca en las filas de detalle.
DetailKeyField		La columna del origen que contiene la clave maestra.
Lines	8	La altura de una banda en filas de la cuadrícula, encabezado anidado incluido. El valor mínimo es 2.
Columns		Las columnas de las cuadrículas de detalle. Vacío usa una copia de las columnas del origen. Solo se almacena cuando tiene columnas.

Una fila muestra un expansor de detalle cuando Active es True, MasterKeyField nombra una columna maestra y hay un Source o un controlador OnGetDetailData asignado. Una fila cuya clave maestra está vacía nunca se expande, y un origen de datos virtual no muestra bandas.

De dónde vienen las filas

- OnGetDetailData se genera primero, sin ningún bloqueo retenido. Sender es la cuadrícula, aRow la fila maestra de la vista, aMasterKey el texto de MasterKeyField y aData los datos de la banda. Rellene aData y asigne True a aHandled para conservar lo escrito; entonces no se lee el Source. Sin Source, las columnas de la banda proceden de Detail.Columns.
- En caso contrario la banda se rellena desde el Source: las filas almacenadas del origen cuyo DetailKeyField coincide con la clave maestra, sea cual sea el filtro y la ordenación del origen. Las filas se copian por FieldName con sus claves de origen. Una columna de detalle sin campo de origen es Null, y un origen virtual no da filas.
- Las claves se comparan como texto, distinguiendo mayúsculas. Los enteros y los números reales sin fracción usan su texto entero, de modo que 5, 5.0 y '5' son la misma clave.
- Los datos de la banda toman el FormatSettings del origen, o el de la cuadrícula maestra cuando las filas proceden solo de OnGetDetailData.

Cuando el origen cambia, cada banda abierta se rellena de nuevo en el fotograma siguiente. Las filas se actualizan en su sitio, de modo que una celda modificada parpadea como en la cuadrícula maestra. OnGetDetailData se genera de nuevo en cada rellenado.

Expandir y contraer

ExpandDetail(aRow) y CollapseDetail(aRow) abren y cierran la banda de la fila maestra de la vista aRow, DetailExpanded(aRow) indica su estado, y OnDetailExpandChange indica la fila y su nuevo estado. DetailData(aRow) devuelve los datos de una banda abierta, nil cuando está contraída. Es de solo lectura; bloquée-la mientras la lee. El estado se conserva por clave maestra, de modo que sobrevive a una ordenación, un filtro, una recarga de las filas maestras y las actualizaciones en vivo.
Un clic en el expansor alterna la banda. Down salta las líneas de una banda. Teclas en una fila seleccionada:

Modo	Teclas
Plano o agrupado, selección de fila	Right y Left expanden y contraen la banda.
Plano o agrupado, SelectionMode gsmCell	Ctrl+Right y Ctrl+Left expanden y contraen la banda; Right y Left siguen moviéndose entre celdas.
Modo árbol	Ctrl+Right y Ctrl+Left expanden y contraen la banda; Right y Left actúan sobre el nodo del árbol.

Dentro de una banda

- La rueda del ratón sobre una banda desplaza la banda. Al final de la banda se desplaza la cuadrícula.
- Un clic en el encabezado anidado ordena las filas de la banda. OnSortChange no se genera.
- Un clic en una celda anidada la selecciona y genera OnDetailCellClick con la fila maestra y la fila de la vista y el índice de Columns de los datos de detalle. La selección maestra no se mueve.
- Las cuadrículas de detalle son de solo lectura y tienen un nivel. SaveToXLSX no escribe las bandas.

En TsgcFMXGrid la rueda desplaza una banda solo mientras el puntero del ratón está sobre ella, un gesto táctil de desplazamiento sigue desplazando la cuadrícula, un toque en el expansor lo alterna, y Cmd funciona como Ctrl.

Ejemplo

Clientes con una banda de sus pedidos, guardados en un TsgcGridDataStore rellenado por código:

```
uses
  SysUtils, Classes,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_Detail, sgcGrid_VCL;

type
  TDetailHandlers = class
  public
    procedure DetailExpandChange(Sender: TObject; aRow: Integer;
      aExpanded: Boolean);
    procedure DetailCellClick(Sender: TObject;
      aRow, aDetailRow, aDetailCol: Integer);
  end;

procedure TDetailHandlers.DetailExpandChange(Sender: TObject; aRow: Integer;
  aExpanded: Boolean);
begin
  // aRow is the master view row whose band was opened or closed
end;

procedure TDetailHandlers.DetailCellClick(Sender: TObject;
  aRow, aDetailRow, aDetailCol: Integer);
var
  oData: TsgcGridData;
  vText: string;
begin
  // aDetailRow and aDetailCol address the data of the band of aRow
  oData := TsgcVCLGrid(Sender).DetailData(aRow);
  if oData = nil then
```

```
    Exit;
  oData.Lock;
  try
    vText := oData.CellText[aDetailRow, aDetailCol];
  finally
    oData.Unlock;
  end;
end;

procedure ShowOrders(Grid: TsgcVCLGrid; Orders: TsgcGridDataStore;
  Handlers: TDetailHandlers);
begin
  // the detail rows: every order names its customer
  Orders.Columns.Add('OrderID', 'Order', gdtInteger);
  Orders.Columns.Add('CustomerID', 'Customer', gdtInteger).Visible := False;
  Orders.Columns.Add('Total', 'Total', gdtCurrency);
  Orders.Data.BeginUpdate;
  try
    Orders.Data.AppendRow('1001', [1001, 1, 250.5]);
    Orders.Data.AppendRow('1002', [1002, 1, 99.9]);
    Orders.Data.AppendRow('1003', [1003, 2, 1200]);
  finally
    Orders.Data.EndUpdate;
  end;
  Grid.OnDetailExpandChange := Handlers.DetailExpandChange;
  Grid.OnDetailCellClick := Handlers.DetailCellClick;
  // the master grid has a CustomerID column
  Grid.Detail.Source := Orders;
  Grid.Detail.MasterKeyField := 'CustomerID';
  Grid.Detail.DetailKeyField := 'CustomerID';
  // band height in grid rows, nested header included
  Grid.Detail.Lines := 6;
  Grid.Detail.Active := True;
  // open the band of the first view row
  Grid.ExpandDetail(0);
end;
```

Las mismas bandas con las filas escritas por OnGetDetailData, por ejemplo desde una consulta por cliente:

```
uses
  SysUtils, Classes,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_VCL;

type
  TOrderLoader = class
  public
    procedure GetDetailData(Sender: TObject; aRow: Integer;
      const aMasterKey: string; aData: TsgcGridData; var aHandled: Boolean);
  end;

procedure TOrderLoader.GetDetailData(Sender: TObject; aRow: Integer;
  const aMasterKey: string; aData: TsgcGridData; var aHandled: Boolean);
begin
  // raised with no lock held; aMasterKey is the MasterKeyField text
  aData.BeginUpdate;
  try
    aData.Clear;
    aData.AppendRow(aMasterKey + '-1', [aMasterKey + '-1', 'Shipped']);
    aData.AppendRow(aMasterKey + '-2', [aMasterKey + '-2', 'Open']);
  finally
    aData.EndUpdate;
  end;
  // the band keeps these rows, the Source is not read
  aHandled := True;
end;

procedure ShowLoadedOrders(Grid: TsgcVCLGrid; Loader: TOrderLoader);
```

```
begin
  // with no Source the bands take their columns from Detail.Columns
  Grid.Detail.Columns.Clear;
  Grid.Detail.Columns.Add('Order', 'Order');
  Grid.Detail.Columns.Add('Status', 'Status');
  Grid.Detail.MasterKeyField := 'CustomerID';
  Grid.Detail.Lines := 5;
  Grid.OnGetDetailData := Loader.GetDetailData;
  Grid.Detail.Active := True;
end;
```

Para mostrar los pedidos de un TDataSet, cárguelo en un TsgcGridDataStore con un TsgcGridSource_DataSet, vea TsgcGridDataStore.

Consulte también

- TsgcGridDataStore
- Filas de árbol
- Enlace con DataSet
- TsgcVCLGrid
- Ediciones
- Filas de árbol y maestro detalle de .NET

Selección y teclado

SelectionMode gsmRow selecciona filas completas y gsmCell celdas individuales. SelectedRow y SelectedCol son la fila de vista y el índice de columna, y SelectedKey la clave de la fila seleccionada: cuando una actualización en vivo mueve la fila, la selección la sigue.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupSelection(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
begin
  Grid.SelectionMode := gsmCell;
  Grid.OnSelectionChange := Handlers.SelectionChange;
  Grid.OnCellClick := Handlers.CellClick;
  // select a row by its key and scroll it into view
  Grid.SelectedRow := Grid.Data.RowOfKey('MSFT');
  if Grid.SelectedRow >= 0 then
    Grid.ScrollToRow(Grid.SelectedRow);
end;
```

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure TGridHandlers.SelectionChange(Sender: TObject);
var
  oGrid: TsgcVCLGrid;
begin
  oGrid := TsgcVCLGrid(Sender);
  if oGrid.SelectedRow >= 0 then
    TForm(oGrid.Owner).Caption := oGrid.SelectedKey + ' ' +
      oGrid.Data.CellText[oGrid.SelectedRow, 1];
end;
```

Teclado

Tecla	Acción
Arriba, Abajo	Fila anterior o siguiente.
Re Pág, Av Pág	Una página arriba o abajo.
Inicio, Fin	Primera o última fila; en modo celda, primera o última columna, y Ctrl+Inicio o Ctrl+Fin la primera o la última fila.
Left, Right	Columna anterior o siguiente en modo celda.
Ctrl+C, Ctrl+Insert	Copia la fila seleccionada como texto separado por tabuladores, o la celda seleccionada en modo celda. Cmd+C en macOS.
F2, Intro, un carácter, Espacio	Inician la edición cuando la cuadrícula no es ReadOnly. Consulte Edición en el lugar.

Ratón

Un clic selecciona, OnCellClick y OnCellDblClick notifican la celda, la rueda desplaza, y RowAt y ColAt localizan la celda situada bajo un punto.

Consulte también

- TsgcVCLGrid
- Cambio de tamaño de columnas

Cambio de tamaño de columnas

Arrastre el borde derecho de una celda de encabezado para cambiar el tamaño de la columna; el cursor cambia sobre el borde. El ancho nunca baja del MinWidth de la columna, 20 de forma predeterminada. El nuevo ancho se guarda en el Width de la columna y se lanza OnColumnResized.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupResizing(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
begin
  Grid.Columns.FindByField('Symbol').MinWidth := 60;
  Grid.OnColumnResized := Handlers.ColumnResized;
end;
```

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure TGridHandlers.ColumnResized(Sender: TObject; aCol: Integer);
begin
  // the new width is already in the column
  OutputDebugString(PChar(Format('%s is now %d wide',
    [TsgcVCLGrid(Sender).Columns[aCol].FieldName,
    TsgcVCLGrid(Sender).Columns[aCol].Width])));
end;
```

Consulte también

- TsgcGridColumn
- Selección y teclado

Columnas fijas

FrozenColumns (0 por defecto) mantiene en su sitio las primeras columnas visibles mientras las demás columnas se desplazan horizontalmente, de modo que una columna clave, como un símbolo o un nombre, siempre está a la vista. El encabezado, el pie y los resúmenes de grupo siguen la misma división, y una línea de borde de un píxel marca el borde fijo. Las columnas fijas forman parte de todas las ediciones, incluida Community.

- FrozenColumns cuenta columnas visibles: una columna oculta se omite.
- Los chevrons de grupo de la izquierda (el margen, un GroupIndent por nivel agrupado) tampoco se desplazan nunca.
- En TsgcVCLGrid la barra de desplazamiento horizontal nativa abarca solo la parte que se desplaza, de modo que el mayor ScrollX no cambia. TsgcFMXGrid dibuja su control deslizante horizontal sobre la parte que se desplaza de la misma manera.
- Los movimientos con el teclado traen a la vista una columna desplazada situada a la derecha del borde fijo.

El renderizador expone el mismo ajuste en sus Options, de modo que la salida PNG y SVG sin interfaz coincide con el control, y FrozenWidth devuelve el ancho que no se desplaza.

Ejemplo

```
uses
  SysUtils, Classes, Forms, Controls,
  sgcGrid_Columns, sgcGrid_Data, sgcGrid_Render, sgcGrid_Lines, sgcGrid_VCL;

procedure FreezeKeyColumns(Grid: TsgcVCLGrid);
begin
  // the first two visible columns stay in place while the others scroll
  Grid.FrozenColumns := 2;
end;

procedure SaveFrozenPNG(Data: TsgcGridData; const aFileName: string);
var
  oRenderer: TsgcGridRenderer;
  oLines: TsgcGridLineView;
begin
  oRenderer := TsgcGridRenderer.Create;
  oLines := TsgcGridLineView.Create(Data);
  try
    oRenderer.Options.FrozenColumns := 1;
    oRenderer.Options.ShowFooter := True;
    oRenderer.SaveToPNG(Data, oLines, sgcGridRenderState, aFileName, 800, 400);
  finally
    oLines.Free;
    oRenderer.Free;
  end;
end;
```

Consulte también

- Cambio de tamaño de columnas
- Agrupación
- Renderizado sin interfaz
- TsgcGridRenderer

Destello al cambiar

Con **FlashChanges** en True (el valor predeterminado), una celda cuyo valor cambió se tiñe con el color de su dirección: FlashUpColor cuando el valor subió, FlashDownColor cuando bajó y FlashChangedColor para cualquier otro cambio. El tinte se desvanece linealmente durante **FlashDuration** milisegundos, 800 de forma predeterminada.
FlashMode fmBackground tiñe el fondo de la celda y fmText el texto. Un color 0 usa el predeterminado del tema: verde, rojo y el acento, translúcidos para el fondo y colores de texto legibles para fmText, más claros en un tema oscuro.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupFlash(Grid: TsgcVCLGrid);
begin
  Grid.FlashChanges := True;
  Grid.FlashDuration := 1200; // ms until the tint has faded out
  Grid.FlashMode := fmBackground;
  // 0 keeps the theme colours; an explicit colour is ARGB
  Grid.FlashUpColor := sgcUIARGB($60, $00, $C8, $50);
  Grid.FlashDownColor := sgcUIARGB($60, $E0, $30, $30);
end;
```

Mientras un destello se desvanece, la cuadrícula sigue solicitando fotogramas, hasta MaxFPS. La selección tiene prioridad sobre el destello, y el destello sobre las reglas de formato.

Consulte también

- Formato condicional
- Temas

Formato condicional

Las **FormatRules** de una cuadrícula colorean las celdas cuyo valor cumple una condición. Una regla compara el valor de la columna FieldName con Value, y con Value2 para gfoBetween, según el DataType de esa columna: los números numéricamente, las fechas como fechas (Value en ISO 8601 o un número de serie de fecha), los booleanos como true o false, el texto sin distinguir mayúsculas de minúsculas. gfoContains y gfoStartsWith comparan siempre texto, gfoIsEmpty coincide con un valor nulo o en blanco, y una celda vacía no coincide con ningún operador de orden.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupRules(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
var
  oRule: TsgcGridFormatRule;
begin
  // red text for a negative change
  Grid.FormatRules.Add('Change', gfoLess, '0', 0,
    sgcUIARGB($FF, $D0, $30, $30));
  // the whole row tinted when a symbol is halted
  oRule := Grid.FormatRules.Add('Halted', gfoEqual, 'true',
    sgcUIARGB($40, $F0, $A0, $20));
  oRule.ApplyToRow := True;
  // bold prices between 100 and 200, and no later rule for that cell
  oRule := Grid.FormatRules.Add;
  oRule.FieldName := 'Price';
  oRule.Operator := gfoBetween;
  oRule.Value := '100';
  oRule.Value2 := '200';
  oRule.Bold := True;
  oRule.StopOnMatch := True;
  // the last word, per painted cell
  Grid.OnGetCellStyle := Handlers.GetCellStyle;
end;
```

Precedencia

- Las reglas se evalúan en el orden de la colección. Cada regla habilitada que coincide aplica sus BackColor y ForeColor distintos de cero, y una coincidencia posterior prevalece sobre una anterior; Bold se acumula.
- Una regla coincidente con StopOnMatch termina la evaluación de esa celda.
- Una regla ApplyToRow evalúa su propia columna y, si coincide, aplica estilo a todas las celdas de la fila.
- Cuando la cuadrícula se dibuja, la selección prevalece sobre el destello, el destello sobre las reglas y las reglas sobre el sombreado de las filas alternas.
- OnGetCellStyle se lanza el último para cada celda dibujada: aFore y aBold llevan el resultado hasta ese momento, y aBack es un relleno adicional encima.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure TGridHandlers.GetCellStyle(Sender: TObject; aRow, aCol: Integer;
  var aBack, aFore: TsgcUIColor; var aBold: Boolean);
var
  oRenderer: TsgcGridRenderer;
```

```
begin
  // Sender is the renderer; aBack is an extra fill over the rule and flash
  oRenderer := TsgcGridRenderer(Sender);
  if (aCol = 0) and (oRenderer.CurrentData.KeyOf(aRow) = 'MSFT') then
    aBold := True;
end;
```

Consulte también

- TsgcGridFormatRule
- Destello al cambiar
- Pregunte a su cuadrícula

Edición en el lugar

La edición en el lugar necesita SGC_GRID_EDIT, que la edición Community no define. Consulte Ediciones.

Iniciar una edición

Establezca **ReadOnly** en False. Una columna es editable mientras su Editable sea True. F2, Intro, un carácter escrito (que reemplaza el texto) o un doble clic abren un TEdit sobre la celda, o un TComboBox cuando la columna tiene un PickList. StartEdit lo abre desde código, y OnBeforeEdit puede rechazarlo.

Confirmar y cancelar

Intro confirma, Tab y Mayús+Tab confirman y pasan a la celda editable siguiente o anterior, y Esc cancela. Perder el foco confirma, o cancela si el texto no es válido. CommitEdit y CancelEdit hacen lo mismo desde código.

Conversión y validación

El texto se convierte al DataType de la columna con Data.FormatSettings. Un texto que no se convierte mantiene abierto el editor y lanza OnEditError, cuyo aMessage se puede modificar. OnValidate recibe el valor convertido y puede modificarlo o rechazarlo. Después el valor se escribe con Data.UpdateCell, de modo que se aplican el seguimiento de cambios, el destello y la ordenación, y OnEdited notifica el valor anterior y el nuevo.

Booleanos

Una columna gdtBoolean no tiene editor: Espacio, un clic, F2, Intro o StartEdit alternan el valor.

Actualizaciones en vivo durante la edición

Cuando una actualización en vivo reordena o filtra la vista, el editor se mueve con su fila, que se vuelve a localizar por su clave. Cuando la fila se elimina o queda filtrada, la edición se cancela.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupEditing(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
begin
  Grid.Columns.Clear;
  Grid.Columns.Add('Item', 'Item').Editable := False;
  Grid.Columns.Add('Qty', 'Qty', gdtInteger);
  Grid.Columns.Add('Side', 'Side').PickList.CommaText := 'Buy,Sell';
  Grid.Columns.Add('Urgent', 'Urgent', gdtBoolean);
  Grid.Data.AppendRow('1', ['Widget', 10, 'Buy', False]);

  Grid.ReadOnly := False;
  Grid.OnValidate := Handlers.Validate;
  Grid.OnEdited := Handlers.Edited;
  Grid.OnEditError := Handlers.EditError;
  // open the editor on the Qty cell of the first row from code
```

```
  Grid.StartEdit(0, 1);
end;

uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure TGridHandlers.Validate(Sender: TObject; aRow, aCol: Integer;
  var aValue: Variant; var aAccept: Boolean);
begin
  // aValue is already converted to the DataType of the column
  if TsgcVCLGrid(Sender).Columns[aCol].FieldName = 'Qty' then
    aAccept := aValue >= 0;
end;

procedure TGridHandlers.Edited(Sender: TObject; aRow, aCol: Integer;
  const aOldValue, aNewValue: Variant);
begin
  OutputDebugString(PChar(VarToStr(aOldValue) + ' -> ' + VarToStr(aNewValue)));
end;

procedure TGridHandlers.EditError(Sender: TObject; aRow, aCol: Integer;
  const aText: string; var aMessage: string);
begin
  aMessage := 'Not a valid quantity: ' + aText;
end;
```

Consulte también

- TsgcGridColumn
- TsgcVCLGrid

Modo virtual

En modo virtual la cuadrícula no almacena filas. **VirtualRowCount** establece cuántas filas hay, y **OnGetValue** de los datos devuelve el valor de una fila y una columna cuando el renderizador las dibuja. Como solo se dibujan las filas visibles, una cuadrícula de un millón de filas cuesta lo mismo que una de cincuenta.
Las filas virtuales no tienen claves, por lo que la vista son las filas en orden: no hay ordenación, filtrado ni seguimiento de cambios, y KeyOf y RowOfKey no devuelven nada. Cambie VirtualRowCount cuando cambie el número de filas.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure ShowVirtual(Grid: TsgcVCLGrid; Handlers: TGridHandlers);
begin
  Grid.Columns.Clear;
  Grid.Columns.Add('Name', 'Name');
  Grid.Columns.Add('Value', 'Value', gdtFloat);
  Grid.Data.OnGetValue := Handlers.GetValue;
  Grid.Data.VirtualRowCount := 1000000;
  Grid.Data.Virtual := True;
end;
```

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure TGridHandlers.GetValue(Sender: TObject; aRow, aCol: Integer;
  var aValue: Variant);
begin
  case aCol of
    0: aValue := 'Row ' + IntToStr(aRow);
    1: aValue := aRow * 1.5;
  end;
end;
```

Consulte también

- TsgcGridData
- Rendimiento

JSON y CSV

La unidad sgcGrid_IO carga y guarda un TsgcGridData como JSON o CSV, desde y hacia cadenas, flujos y archivos. Es independiente del framework y también se compila en Linux64.

Carga

- Una carga reemplaza las filas y se ejecuta dentro de un único BeginUpdate y EndUpdate. Con un campo clave, la clave de una fila es el texto de ese campo; una clave repetida actualiza la fila anterior.
- Con aAutoColumns en True y sin columnas todavía, las columnas se crean a partir de la entrada y su DataType se deduce de los valores. Las columnas existentes se conservan y se emparejan por FieldName.
- La entrada JSON es una matriz de objetos. Un FieldName con puntos, como price.last, lee miembros anidados.
- CSV sigue la RFC 4180, y el primer registro contiene los FieldNames. aInvariant en True (el valor predeterminado) lee y escribe los números con punto y las fechas en ISO 8601; False usa los FormatSettings de los datos.

Guardado

GridSaveToJSON y GridSaveToCSV escriben las filas en el orden de la vista, ordenadas y filtradas, con las columnas en orden, solo las visibles cuando aVisibleOnly es True. El texto es UTF-8 sin BOM.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure ExportImport(Grid: TsgcVCLGrid);
var
  vJSON: string;
begin
  // the rows in view order, as an array of objects
  vJSON := GridSaveToJSON(Grid.Data);
  // the visible columns only, invariant numbers and ISO dates
  GridSaveToCSVFile(Grid.Data, 'quotes.csv', True);

  // a load replaces the rows; the key of a row is its Symbol
  GridLoadFromJSON(Grid.Data,
    '[{"Symbol":"MSFT","Price":412.5},{"Symbol":"AAPL","Price":228.1}]',
    'Symbol');
  // semicolon delimited, numbers in the FormatSettings of the data
  GridLoadFromCSVFile(Grid.Data, 'quotes.csv', 'Symbol', True, ';', False);
end;
```

Consulte también

- Enlace con DataSet
- TsgcGridData

Exportación a Excel

SaveToXLSX escribe la cuadrícula como un libro de Excel (Office Open XML, una hoja de cálculo) en un flujo o un archivo, sin ninguna biblioteca externa. El libro conserva lo que muestra la cuadrícula: las columnas visibles en el orden de visualización con su ancho y sus formatos numéricos actuales, el encabezado fijo y FrozenColumns, los grupos o el árbol como un esquema contraíble, y los resúmenes como fórmulas.
Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.

Exportar una cuadrícula

Método	Descripción
SaveToXLSX(aStream; aSheetName = 'Sheet1')	Escribe el libro en un flujo.
SaveToXLSX(aFileName; aSheetName = 'Sheet1')	Escribe el libro en un archivo.

Llámelo en el hilo de la interfaz. El nombre de la hoja pierde los caracteres que Excel rechaza (se convierten en _) y se corta a 31 caracteres. Una hoja que necesite más de 1.048.576 filas o 16.384 columnas genera EsgcGridXLSXError antes de escribir nada.

Qué contiene la hoja

- La fila 1 es el encabezado en negrita, fijado junto con las primeras FrozenColumns columnas. Los anchos de columna son los anchos en pantalla.
- Se escribe cada fila de grupo y cada fila de la vista, esté lo que esté contraído. Las filas de un grupo contraído se ocultan y la fila de grupo se marca como contraída, de modo que Excel abre el esquema tal como lo muestra la cuadrícula. Las filas de grupo van encima de sus filas.
- Una fila de grupo contiene su título (Region: North (12)) en la primera columna, o en la primera columna sin resumen. OnGetGroupText no se usa.
- Las filas de árbol reciben el nivel de esquema de su profundidad y una sangría en la columna del árbol. Los niveles de esquema se detienen en 7, el máximo de Excel.
- Las bandas de detalle no se exportan.
- Los números se escriben como números, las fechas como fechas seriales de Excel, los booleanos como booleanos y el texto como cadenas compartidas sin caracteres de control.
- El DisplayFormat de una columna se convierte en el formato numérico de Excel. Las máscaras de FormatFloat pasan tal cual; en las máscaras de fecha nn se convierte en mm, z en 000 y am/pm en AM/PM. Sin DisplayFormat, una columna de enteros usa 0, una columna de moneda #,##0.00 y una columna de fecha yyyy-mm-dd hh:mm:ss.

Resúmenes

Cada columna de resumen de una fila de grupo contiene una fórmula SUBTOTAL sobre las filas del grupo, con el valor de la cuadrícula en caché en la celda, de modo que el archivo muestra los totales antes de que Excel recalcule. Cuando una columna visible tiene un resumen, una fila de pie contiene SUBTOTAL sobre todas las filas de datos. La fila de pie se escribe sea cual sea el valor de ShowFooter.

SummaryType	SUBTOTAL	Tipos de columna
gsuSum	9	Números
gsuAverage	1	Números
gsuMin	5	Números y fechas

gsuMax	4	Números y fechas
gsuCount	3	Cualquier tipo (celdas no vacías)

Estos códigos cuentan las filas ocultas, de modo que un grupo contraído conserva su total, y SUBTOTAL ignora las celdas SUBTOTAL anidadas de los grupos interiores.

TsgcGridXLSXWriter

El escritor que hay detrás de SaveToXLSX está en la unidad sgcGrid_XLSX y no necesita ninguna cuadrícula: escribe cualquier TsgcGridData.

Miembro	Descripción
SaveToStream(aData, aLines, aStream), SaveToFile(aData, aLines, aFileName)	Escribe el libro de aData. aLines da la agrupación, el árbol y el estado expandido; nil usa una vista de líneas propia.
SheetName	El nombre de la hoja de cálculo, 'Sheet1' por defecto.
FrozenColumns	Las primeras columnas visibles que se mantienen a la vista.
SetColumnWidth(aCol, aDIPs)	El ancho de la columna aCol (índice de Columns) en DIP, usado en lugar de su Width. Un valor negativo restaura Width.
Summaries	Escribe las celdas SUBTOTAL de grupo y de pie, True por defecto.
FileTime	La fecha y la hora de las entradas del zip, Now cuando se crea el escritor. Los mismos datos y el mismo FileTime dan siempre los mismos bytes.
sgcGridExcelNumberFormat(aDataType, aDisplayFormat)	El código de formato numérico de Excel de un tipo de columna y un DisplayFormat.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcGrid_Data, sgcGrid_XLSX, sgcGrid_VCL;

function ExportGrid(Grid: TsgcVCLGrid; const aFileName: string): Boolean;
begin
  // the visible columns with their current widths, FrozenColumns, the
  // groups or the tree as an outline and the summaries
  Result := True;
  try
    Grid.SaveToXLSX(aFileName, 'Orders');
  except
    on E: EsgcGridXLSXError do
      // too many rows or columns, nothing was written
      Result := False;
  end;
end;

procedure ExportToStream(Grid: TsgcVCLGrid; aStream: TStream);
begin
```

```
  // the sheet is named Sheet1
  Grid.SaveToXLSX(aStream);
end;

procedure ExportData(aData: TsgcGridData; aStream: TStream);
var
  oWriter: TsgcGridXLSXWriter;
begin
  // headless: no grid needed
  oWriter := TsgcGridXLSXWriter.Create;
  try
    oWriter.SheetName := 'Prices';
    oWriter.FrozenColumns := 1;
    // no SUBTOTAL rows
    oWriter.Summaries := False;
    // a fixed time gives the same bytes for the same rows
    oWriter.FileTime := EncodeDate(2026, 1, 1);
    // first column 160 DIPs wide instead of its Width
    oWriter.SetColumnWidth(0, 160);
    // nil: the writer uses a line view of its own
    oWriter.SaveToStream(aData, nil, aStream);
  finally
    oWriter.Free;
  end;
end;
```

Consulte también

- JSON y CSV
- Resúmenes
- Agrupación
- Filas de árbol
- Columnas fijas
- Ediciones
- Exportación a Excel de .NET

Enlace con DataSet

Un TsgcGridSource_DataSet rellena una cuadrícula desde cualquier TDataSet. Con un KeyField la cuadrícula se sincroniza en lugar de recargarse: las filas modificadas se actualizan en el lugar y destellan, las claves nuevas se añaden y las que faltan se eliminan. Con LiveUpdates la cuadrícula sigue cada cambio confirmado del conjunto de datos.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

function BindDataSet(Grid: TsgcVCLGrid; DataSet: TDataSet)
  : TsgcGridSource_DataSet;
begin
  Result := TsgcGridSource_DataSet.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.DataSet := DataSet;
  Result.KeyField := 'ID'; // rows are updated in place by ID
  Result.AutoColumns := True; // columns from the fields when the grid has none
  Result.LiveUpdates := True; // follow Post, Delete and Refresh
  Result.Active := True;
end;
```

El conjunto de datos se recorre con DisableControls y se restaura su registro actual. Cerrar el conjunto de datos conserva las filas. La demo DataSetGrid muestra el enlace con un TClientDataSet.

Consulte también

- TsgcGridSource_DataSet
- JSON y CSV
- Demos

Flujos en vivo

Los orígenes de flujo de sgcGridWS escriben mensajes JSON de los clientes de sgcWebSockets directamente en una cuadrícula: TsgcGridFeedSource_WebSocket, TsgcGridFeedSource_MQTT y TsgcGridFeedSource_SSE. Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Reglas de asignación

Propiedad	Regla
RowsPath	Dónde están las filas en un mensaje. Vacío: el mensaje es un objeto fila o una matriz de filas.
KeyPath	La ruta de la clave dentro de una fila, como symbol o id.value. Se acepta una cadena, un número o un booleano.
Fields	Líneas FieldName=json.path; una línea sin = lee la ruta del mismo nombre; vacío lee cada columna de la ruta con el nombre de su FieldName. Solo se escriben las rutas presentes, de modo que las actualizaciones parciales funcionan.
DeletePath	Una ruta booleana que elimina la fila de la clave cuando es true.

WebSocket

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function FeedFromWebSocket(Grid: TsgcVCLGrid; Client: TsgcWebSocketClient)
  : TsgcGridFeedSource_WebSocket;
begin
  // messages such as {"quotes":[{"symbol":"MSFT","quote":{"last":412.5}}]}
  Result := TsgcGridFeedSource_WebSocket.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.RowsPath := 'quotes';
  Result.KeyPath := 'symbol';
  Result.Fields.Add('Symbol=symbol');
  Result.Fields.Add('Price=quote.last');
  Result.Fields.Add('Volume'); // same-named path
  Result.DeletePath := 'removed';
  Result.Client := Client;
  Client.URL := 'wss://quotes.example.com/stream';
  Client.Active := True;
end;
```

MQTT

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;
```

```
function FeedFromMQTT(Grid: TsgcVCLGrid; MQTT: TsgcWSPClient_MQTT)
  : TsgcGridFeedSource_MQTT;
begin
  Result := TsgcGridFeedSource_MQTT.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.KeyPath := 'id';
  Result.Topic := 'plant/+/sensors';
  Result.Client := MQTT;
  // the source does not subscribe: subscribe on the MQTT client, for
  // example in its OnMQTTConnect event
end;
```

Eventos enviados por el servidor

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function FeedFromSSE(Grid: TsgcVCLGrid; HTTP: TsgcHTTP1Client)
  : TsgcGridFeedSource_SSE;
begin
  Result := TsgcGridFeedSource_SSE.Create(Grid.Owner);
  Result.Grid := Grid;
  Result.KeyPath := 'orderId';
  Result.Client := HTTP;
  // the application starts the stream
  HTTP.GetSSE('https://api.example.com/orders/stream');
end;
```

Mensajes de otro origen

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

procedure FeedByHand(Source: TsgcGridFeedSource_WebSocket);
begin
  // apply a message that did not come from the client, returns the rows
  Source.ProcessMessage('[{"symbol":"MSFT","Price":413.2},' +
    '{"symbol":"IBM","removed":true}]');
end;
```

Las filas de un mensaje se aplican en un único BeginUpdate y EndUpdate en el hilo del cliente, lo cual es seguro porque los datos son seguros para subprocesos. La demo LiveFeed muestra el origen WebSocket con un servidor en el mismo proceso.

Consulte también

- TsgcGridFeedSource
- Actualizaciones en vivo y subprocesos

Pregunte a su cuadrícula

TsgcGridAIQuery permite que un usuario filtre, ordene y resalte una cuadrícula con una pregunta en lenguaje natural, como price greater than 100 sort by volume descending highlight change less than 0.
Esta característica forma parte del puente sgcGridWS: las unidades necesitan las fuentes o los paquetes de sgcWebSockets, y la edición Community no la incluye.

Cómo se aplica la respuesta

- Los filtros se combinan con AND y se aplican mediante OnFilterRow de los datos. Un controlador ya asignado se conserva y se llama primero.
- Las ordenaciones sustituyen SortOrder y SortIndex de las columnas.
- Los resaltados se añaden a FormatRules de la cuadrícula con ApplyToRow activado.
- La agrupación y los límites de filas no son compatibles con la cuadrícula; se ignoran y se notifican en LastExplanation.
- Cada Ask llama primero a Reset, de modo que las preguntas no se acumulan. Reset restaura el controlador de filtro y la ordenación anteriores y elimina solo las reglas que añadió la consulta.

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

function AskGrid(Grid: TsgcVCLGrid): string;
var
  oQuery: TsgcGridAIQuery;
begin
  oQuery := TsgcGridAIQuery.Create(Grid.Owner);
  oQuery.Grid := Grid;
  // no Assistant: the rule-based parser answers, no network, no API key
  if oQuery.Ask('price greater than 100 sort by volume descending ' +
    'highlight change less than 0') then
    Result := oQuery.LastExplanation
  else
    Result := 'not understood';
  // later: remove the filter, the sort and the highlight
  oQuery.Reset;
end;
```

Con un LLM

Asigne un TsgcHTMLAIAssistant con un proveedor para responder con un LLM. Ask es síncrono y se bloquea mientras responde el proveedor, por lo que debe llamarlo desde el hilo principal. Sin Assistant responde el analizador basado en reglas, sin red y sin clave de API, como en la demo AskYourGrid.

```
uses
  Windows, SysUtils, Classes, Forms,
  sgcWebSocket, sgcWebSocket_Protocols, sgcHTTP, sgcHTML_AI,
  sgcGrid_Columns, sgcGrid_VCL, sgcGridWS_Source, sgcGridWS_Source_WebSocket,
  sgcGridWS_Source_MQTT, sgcGridWS_Source_SSE, sgcGridWS_AI;

procedure UseAssistant(Query: TsgcGridAIQuery; Assistant: TsgcHTMLAIAssistant);
begin
  // an Assistant with a provider answers with an LLM; Ask blocks meanwhile
  Query.Assistant := Assistant;
  Query.Ask('show the halted symbols with the biggest loss first');
end;
```

Consulte también

- TsgcGridAIQuery
- Filtrado
- Formato condicional

Gráficos desde la cuadrícula

Los datos de una cuadrícula implementan IsgcUITable, la interfaz de tabla de sgcUICore, y las cuadrículas la delegan. El origen de datos de sgcCharts **TsgcChartDataSource_Table** lee una tabla así: establezca su Table en la cuadrícula, y XColumn e YColumn en FieldNames de columna. Cada cambio de las filas de la cuadrícula, ya sea una actualización de un flujo o una edición en el lugar, llega al gráfico sin código intermedio, y el gráfico sigue la ordenación y el filtro de la cuadrícula.

```
uses
  Forms, Classes, Controls,
  sgcGrid_Columns, sgcGrid_VCL,
  sgcCharts_Chart, sgcCharts_Series, sgcCharts_DataSource_Table, sgcCharts_VCL;

procedure ChartFromGrid(Form: TForm; Grid: TsgcVCLGrid);
var
  oChart: TsgcChart;
  oSeries: TsgcChartSeries_Line;
  oSource: TsgcChartDataSource_Table;
  oView: TsgcVCLChart;
begin
  Grid.Columns.Add('x', 'x', gdtFloat);
  Grid.Columns.Add('y', 'y', gdtFloat);
  Grid.Data.AppendRow('1', [1, 10]);
  Grid.Data.AppendRow('2', [2, 14]);

  oChart := TsgcChart.Create(Form);
  oSeries := TsgcChartSeries_Line.Create(Form);
  oSeries.Chart := oChart;
  oSource := TsgcChartDataSource_Table.Create(Form);
  oSource.Series := oSeries;
  oSource.XColumn := 'x'; // a column FieldName
  oSource.YColumn := 'y';
  oSource.Table := Grid; // the grid implements IsgcUITable

  oView := TsgcVCLChart.Create(Form);
  oView.Parent := Form;
  oView.Align := alRight;
  oView.Chart := oChart;

  // every change of the grid rows now reaches the chart
  Grid.Data.UpdateRow('2', ['y'], [18]);
end;
```

El ejemplo necesita las unidades de sgcCharts. La demo GridAndChart coloca la cuadrícula y el gráfico en dos mosaicos de sgcDashboard.

Consulte también

- IsgcGridDataHost e IsgcUITable
- Demos

Temas

La cuadrícula usa los tokens de tema claro y oscuro de sgcUICore: Theme utLight (predeterminado) o utDark, AccentColor para la selección y el destello de cambio, y FontSize, RowHeight, HeaderHeight, AlternateRows y las líneas de cuadrícula para la densidad. Los colores son valores ARGB TsgcUIColor; sgcUIARGB construye uno.
Un TsgcVCLGrid colocado en un mosaico de sgcDashboard sigue el tema del panel mediante su propiedad publicada Theme.
Los ejemplos son procedimientos simples que reciben un **TsgcVCLGrid**. Los controladores de eventos son métodos de una pequeña clase, TGridHandlers, declarada en TsgcVCLGrid. El mismo código funciona con TsgcFMXGrid.

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SetupTheme(Grid: TsgcVCLGrid);
begin
  Grid.Theme := utDark;
  Grid.AccentColor := sgcUIARGB($FF, $E0, $80, $20);
  Grid.RowHeight := 24;
  Grid.HeaderHeight := 30;
  Grid.FontSize := 12;
  Grid.AlternateRows := False;
  Grid.GridLinesV := False;
  Grid.MaxFPS := 30; // repaint at most 30 times per second
end;
```

Consulte también

- Destello al cambiar
- Renderizado sin interfaz

Renderizado sin interfaz

El renderizador no necesita una ventana. SaveToPNG y SaveToSVG de las cuadrículas escriben la vista actual; los procedimientos sgcGridSaveToPNG y sgcGridSaveToSVG, o un TsgcGridRenderer, renderizan un TsgcGridData independiente, también en Linux64, para informes, correos electrónicos o imágenes generadas en el servidor. La imagen de la Introducción se hizo con el ejemplo siguiente.

```
uses
  SysUtils,
  sgcUICore_Theme, sgcGrid_Columns, sgcGrid_Data, sgcGrid_Format,
  sgcGrid_Render;

var
  oData: TsgcGridData;
  oOptions: TsgcGridRenderOptions;
  vState: TsgcGridRenderState;
begin
  // nil: the data creates and owns its own columns
  oData := TsgcGridData.Create(nil);
  oOptions := TsgcGridRenderOptions.Create;
  try
    // '.' decimals and ',' thousands whatever the locale
    oData.UseInvariantFormat;
    oData.Columns.Add('Symbol', 'Symbol');
    oData.Columns.Add('Price', 'Price', gdtFloat).DisplayFormat := '#,##0.00';
    oData.Columns.Add('Change', 'Change %', gdtFloat).DisplayFormat :=
      '+0.00;-0.00;0.00';
    oData.Columns.Add('Volume', 'Volume', gdtInteger).DisplayFormat := '#,##0';
    oData.Columns.Add('Halted', 'Halted', gdtBoolean).Width := 70;
    oData.AppendRow('MSFT', ['MSFT', 412.5, 1.25, 1200500, False]);
    oData.AppendRow('AAPL', ['AAPL', 228.1, -0.84, 3400200, False]);
    oData.AppendRow('GOOG', ['GOOG', 171.9, 0.37, 980100, False]);
    oData.AppendRow('AMZN', ['AMZN', 186.4, -2.10, 2250000, True]);
    oData.AppendRow('NVDA', ['NVDA', 121.7, 3.02, 5120750, False]);
    oData.Columns[1].SortOrder := gsoDescending;

    oOptions.Theme := utDark;
    oOptions.FormatRules.Add('Change', gfoLess, '0', 0,
      sgcUIARGB($FF, $F0, $60, $60));
    oOptions.FormatRules.Add('Volume', gfoGreater, '5000000', 0, 0, True);

    vState := sgcGridRenderState;
    vState.SelectedRow := 1;
    // 470 x 200 logical pixels, written at scale 2 as 940 x 400 pixels
    sgcGridSaveToPNG(oData, oOptions, vState, 'grid.png', 470, 200, 2);
    sgcGridSaveToSVG(oData, oOptions, vState, 'grid.svg', 470, 200);
  finally
    oOptions.Free;
    oData.Free;
  end;
end;
```

Desde una cuadrícula

```
uses
  Windows, SysUtils, Classes, Variants, Forms, Controls, DB,
  sgcUICore_Canvas, sgcUICore_Theme, sgcUICore_Table, sgcGrid_Columns,
  sgcGrid_Data, sgcGrid_Format, sgcGrid_Render, sgcGrid_IO,
  sgcGrid_Source_DataSet, sgcGrid_VCL;

procedure SaveGridImage(Grid: TsgcVCLGrid);
begin
```

```
  // the rows of the current view, without a window, 800 x 400 pixels
  Grid.SaveToPNG('grid.png', 800, 400);
  Grid.SaveToSVG('grid.svg', 800, 400);
end;
```

Consulte también

- TsgcGridRenderer
- Temas

Descripción general de .NET

sgcGrid .NET es un port administrado en C# de la biblioteca sgcGrid de Delphi, no un envoltorio alrededor de ella. Las clases de C# están en el espacio de nombres **esegece.sgcGrid** y conservan los nombres de Delphi, de modo que **TsgcGridData**, **TsgcGridColumns**, **TsgcGridColumn**, **TsgcGridFormatRules**, **TsgcGridRenderer** y **TsgcGridSource_DataSet** se leen igual en ambos lenguajes. Los controles de cuadrícula siguen el patrón de nombres de Delphi: **TsgcWinFormsGrid** para WinForms, **TsgcWpfGrid** para WPF y **TsgcAvaloniaGrid** para Avalonia. Los tres dibujan con el port .NET de **sgcUICore**, con el mismo lienzo, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, por lo que una cuadrícula se ve igual en cualquier host y en la salida sin interfaz en PNG y SVG.
La cuadrícula está pensada para datos en vivo. El almacén de filas está indexado por clave y es seguro para subprocesos, así que un subproceso de trabajo puede actualizar, añadir y eliminar filas mientras la cuadrícula dibuja solo las filas visibles a una frecuencia de fotogramas controlada. Las celdas modificadas destellan, las reglas de formato colorean celdas o filas completas, y la cuadrícula ordena, filtra, funciona en modo virtual y exporta a JSON y CSV. En la edición Registered el usuario edita las celdas en el lugar, el puente con sgcWebSockets alimenta la cuadrícula desde conexiones WebSocket, MQTT y SSE, y el puente con sgcHTML responde preguntas en lenguaje natural sobre las filas.

Paquetes

El port se divide en paquetes NuGet, de modo que un proyecto solo incorpora lo que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcGrid	El modelo (**TsgcGridColumns**, **TsgcGridData**, **TsgcGridFormatSettings**), ordenación y filtrado, modo virtual, destello y reglas de formato (**TsgcGridFormatRules**), el renderizador **TsgcGridRenderer** con la salida SVG sin interfaz, importación y exportación de JSON y CSV (**sgcGrid_IO**) y el origen DataTable **TsgcGridSource_DataSet**. Sin dependencia nativa.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGrid.Rendering	La salida PNG sin interfaz (**SaveToPNG**), mediante SkiaSharp.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGrid.WinForms	**TsgcWinFormsGrid**.	net472, net8.0-windows, net9.0-windows
esegece.sgcGrid.Wpf	**TsgcWpfGrid**.	net472, net8.0-windows, net9.0-windows
esegece.sgcGrid.Avalonia	**TsgcAvaloniaGrid**, sobre Avalonia 11.3.	net8.0, net9.0
esegece.sgcGrid.sgcWebSockets	El puente de flujos en vivo: **TsgcGridFeedSource**, **TsgcGridFeedSource_WebSocket**, **TsgcGridFeedSource_MQTT** y	netstandard2.0,

	TsgcGridFeedSource_SSE. Depende del paquete esegece.sgcWebSockets. Solo en la edición Registered.	net8.0, net9.0
esegece.sgcGrid.sgcHTML	El puente de consultas con IA: **TsgcGridAIQuery** y **TsgcGridAIFilterItem**, sobre el **TsgcHTMLAIAssistant** del paquete esegece.sgcHTML. Solo en la edición Registered.	netstandard2.0, net8.0, net9.0

Cada paquete de host depende de esegece.sgcGrid.Rendering y del paquete sgcUICore del mismo host, por lo que basta con añadir el paquete del host.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcGrid, .Rendering, .WinForms, .Wpf, .Avalonia, .sgcWebSockets y .sgcHTML	Todo: las tres cuadrículas, actualizaciones en vivo, destello y reglas de formato, ordenación, filtrado, modo virtual, JSON y CSV, el origen DataTable, la salida sin interfaz, la edición en el lugar y ambos puentes.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcGrid.WinForms.Community. No hay paquetes de puente Community.	Todo salvo la edición en el lugar (**StartEdit**, **CommitEdit**, **CancelEdit**, los eventos **OnBeforeEdit**, **OnValidate**, **OnEdited** y **OnEditError**) y los dos puentes.

Es la misma división que la de las ediciones de Delphi, consulte Ediciones.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	TsgcVCLGrid	Windows, sobre .NET Framework 4.7.2 y .NET 8 y 9.
WPF	TsgcFMXGrid	Windows, sobre .NET Framework 4.7.2 y .NET 8 y 9.
Avalonia	TsgcFMXGrid	Windows, Linux y macOS, sobre .NET 8 y 9.
Headless	TsgcGridRenderer	Cualquier plataforma de .NET Standard 2.0, incluidos los servidores Linux.

Consulte también

- Instalación desde NuGet

- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias entre .NET y Delphi
- Renderizado sin interfaz en .NET
- Enlace con DataTable, JSON y CSV en .NET
- Edición en el lugar en .NET
- Flujos en vivo en .NET
- Pregunte a su cuadrícula en .NET
- Rendimiento en .NET
- Demos de .NET

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, y los paquetes core y de renderizado vienen con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcGrid.WinForms
dotnet add package esegece.sgcGrid.Wpf
dotnet add package esegece.sgcGrid.Avalonia
```

Añada los puentes para los flujos en vivo y las consultas en lenguaje natural. Traen consigo los paquetes esegece.sgcWebSockets y esegece.sgcHTML:

```
dotnet add package esegece.sgcGrid.sgcWebSockets
dotnet add package esegece.sgcGrid.sgcHTML
```

Edición Community

```
dotnet add package esegece.sgcGrid.WinForms.Community
dotnet add package esegece.sgcGrid.Wpf.Community
dotnet add package esegece.sgcGrid.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcGrid.WinForms" Version="2026.10.0" />
</ItemGroup>
```

En un proyecto, haga referencia a los paquetes Registered o a los Community, nunca a ambos: contienen los mismos tipos en el mismo espacio de nombres. El código que usa la API de edición no compila con los paquetes Community; protéjalo con `#if !SGC_COMMUNITY` cuando un mismo proyecto compile ambas ediciones.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto de .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto de .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:grid="clr-namespace:esegece.sgcGrid;assembly=esegece.sgcGrid.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como en cualquier aplicación de escritorio de Avalonia. Asigne el espacio de nombres en XAML con `xmlns:grid="using:esegece.sgcGrid"`.
- Sin interfaz: esegece.sgcGrid por sí solo basta para el modelo, el origen DataTable, JSON, CSV y SVG. Añada esegece.sgcGrid.Rendering para PNG, que trae SkiaSharp y sus recursos nativos para Windows y Linux.
- Puentes: esegece.sgcGrid.sgcWebSockets y esegece.sgcGrid.sgcHTML tienen como destino netstandard2.0, net8.0 y net9.0, por lo que funcionan con todos los hosts, incluido .NET Framework 4.7.2.

En C# todos los hosts necesitan `using esegece.sgcGrid;`, y `using esegece.sgcUICore;` para **TsgcUITheme** y los demás tipos de los temas. Los orígenes de flujos en vivo también necesitan `using esegece.sgcWebSockets;` para los clientes de socket.

Consulte también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF

- Inicio rápido Avalonia
- Flujos en vivo en .NET
- Pregunte a su cuadrícula en .NET
- Demos de .NET

Inicio rápido WinForms

Este ejemplo crea un formulario WinForms con una cuadrícula de cotizaciones. Un subproceso de trabajo actualiza las filas cuatro veces por segundo, las celdas modificadas destellan, una regla de formato pinta en rojo claro las filas con variación negativa y una casilla de verificación cambia la cuadrícula entre el tema claro y el oscuro.

Pasos

- Cree un proyecto WinForms `net8.0-windows` y añada el paquete **esegece.sgcGrid.WinForms**, consulte Instalación desde NuGet.
- Cree un **TsgcWinFormsGrid** y acóplelo. Añada las columnas con **Columns.Add**: el nombre del campo, el título y el **TsgcGridDataType**, y establezca **DisplayFormat** para números y fechas.
- Añada las filas con **Data.AppendRow**: el primer argumento es la clave de la fila y los valores siguen el orden de las columnas.
- Actualice las filas por clave con **Data.UpdateRow** o **Data.UpdateCell** desde cualquier subproceso. **BeginUpdate** y **EndUpdate** agrupan muchos cambios en un solo repintado.
- Añada una regla a **FormatRules** para colorear una celda, o la fila completa con **ApplyToRow**.

Ejemplo

Los colores son valores `$AARRGGBB`; un alfa inferior a `FF` tiñe la celda sobre su fondo. La cuadrícula se repinta como máximo **MaxFPS** veces por segundo, por rápido que cambien los datos, consulte Actualizaciones en vivo y subprocesos y Rendimiento en .NET.

Consulte también

- Instalación desde NuGet
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- TsgcVCLGrid
- Destello al cambiar
- Formato condicional
- Demos de .NET

Inicio rápido WPF

La misma cuadrícula de cotizaciones en WPF. La cuadrícula y sus opciones de visualización se declaran en XAML; las columnas y las filas se establecen desde el código subyacente.

Pasos

- Cree un proyecto WPF `net8.0-windows` y añada el paquete **esegece.sgcGrid.Wpf**, consulte Instalación desde NuGet.
- Asigne el espacio de nombres en la ventana con `xmlns:grid="clr-namespace:esegece.sgcGrid;assembly=esegece.sgcGrid.Wpf"`.
- Declare un **TsgcWpfGrid** y establezca **RowHeight**, **FlashChanges**, **TailMode** y las demás opciones de visualización como atributos.
- Añada las columnas, las reglas de formato y las filas en el constructor de la ventana, después de `InitializeComponent`.

Ejemplo

MainWindow.xaml:
MainWindow.xaml.cs:
App.xaml es el de un proyecto WPF nuevo, con `StartupUri="MainWindow.xaml"`. La cuadrícula WPF dibuja sus propias barras de desplazamiento, como hace la cuadrícula FireMonkey, consulte Diferencias entre .NET y Delphi.

Consulte también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias entre .NET y Delphi
- TsgcFMXGrid
- Demos de .NET

Inicio rápido Avalonia

La misma cuadrícula de cotizaciones en Avalonia, en Windows, Linux y macOS.

Pasos

- Cree un proyecto de escritorio Avalonia `net8.0` con **Avalonia.Desktop** y **Avalonia.Themes.Fluent** 11.3, y añada el paquete **esegece.sgcGrid.Avalonia**, consulte Instalación desde NuGet.
- Asigne el espacio de nombres en la ventana con `xmlns:grid="using:esegece.sgcGrid"`.
- Declare un **TsgcAvaloniaGrid** con un `Name`, búsquelo después de `AvaloniaXamlLoader.Load` y añada las columnas y las filas.
- Cambie **Theme** desde el código subyacente.

Ejemplo

MainWindow.axaml:
MainWindow.axaml.cs:
Program.cs:
En una pantalla táctil el contenido sigue al dedo y un toque selecciona la celda. A través de una referencia de tipo `Control`, **Theme** es el tema de control de Avalonia, consulte Diferencias entre .NET y Delphi.

Consulte también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Clases de .NET
- Diferencias entre .NET y Delphi
- TsgcFMXGrid
- Demos de .NET

Clases de .NET

Todas las clases están en el espacio de nombres **esegece.sgcGrid**. Los tres controles de cuadrícula son hosts delgados sobre el modelo y el renderizador independientes del framework, que comparten los hosts, la salida sin interfaz y los puentes.

Clases de host

Host	Clase	Clase base	Delphi
WinForms	TsgcWinFormsGrid	TsgcUIWinFormsControl	TsgcVCLGrid
WPF	TsgcWpfGrid	TsgcUIWpfControl	TsgcFMXGrid
Avalonia	TsgcAvaloniaGrid	TsgcUIAvaloniaControl	TsgcFMXGrid

Las tres cuadrículas implementan **IsgcUITable**, de modo que un gráfico de sgcCharts puede leerlas mediante **TsgcChartDataSource_Table**.

Tipos independientes del framework

Tipo	Descripción
TsgcGridColumns, TsgcGridColumn	Las columnas: **FieldName**, **Caption**, **DataType**, **Width**, **MinWidth**, **Alignment**, **DisplayFormat**, **Visible**, **Editable**, **SortOrder**, **SortIndex**, **PickList**, además de **GroupIndex**, **SummaryType** y **SummaryFormat** para la agrupación y los resúmenes. Consulte TsgcGridColumn.
TsgcGridData	El almacén de filas indexado por clave y seguro para subprocesos: **AppendRow**, **UpdateRow**, **UpdateCell**, **DeleteRow**, **Clear**, **BeginUpdate** y **EndUpdate**, sellos de cambio, modo tail con **MaxRows**, ordenación, **FilterText** y **OnFilterRow**, modo virtual (**Virtual**, **VirtualRowCount**, **OnGetValue**) y **FormatSettings**. **LayoutVersion** avanza con cada cambio que pueda cambiar la agrupación o el árbol, y **SetLayoutFields** añade campos cuyos cambios de valor también la hacen avanzar. Consulte TsgcGridData.
TsgcGridDataType	`gdtText` , `gdtInteger` , `gdtFloat` , `gdtCurrency` , `gdtDateTime` y `gdtBoolean` .
TsgcGridFormatRules, TsgcGridFormatRule	Formato condicional: **FieldName**, **Operator** (de `gfoEqual` a `gfoIsEmpty`), **Value**, **Value2**, **BackColor**, **ForeColor**, **Bold**, **ApplyToRow**, **Enabled** y **StopOnMatch**. Consulte TsgcGridFormatRule.
TsgcGridSummaryType	`gsuNone` , `gsuSum` , `gsuCount` , `gsuAverage` , `gsuMin` y `gsuMax` .
TsgcGridLineView, TsgcGridGroupInfo	Las líneas de visualización de unos datos: filas de grupo, filas de datos, filas de árbol y bandas de detalle, con **GroupBy**, **GroupCount**, **GroupInfo**, **GroupSummary** y el estado expandido. Consulte Agrupación y resúmenes de .NET.
TsgcGridTreeOptions	Filas de árbol: **KeyField**, **ParentField**, **TreeField**, **Indent** y **Expanded**. Consulte Filas de árbol y maestro detalle de .NET. Solo en la edición Registered.
TsgcGridDetailOptions	Bandas de detalle: **Active**, **Source**, **MasterKeyField**, **DetailKeyField**, **Lines** y **Columns**. Consulte Filas de árbol y maestro detalle de .NET. Solo en la edición Registered.

TsgcGridDataStore	Un contenedor no visual de filas de detalle, que se rellena desde código o con un **TsgcGridSource_DataSet**. Solo en la edición Registered.
TsgcGridXLSXWriter	Escribe unos datos y su vista de líneas como un libro de Excel. Consulte Exportación a Excel de .NET. Solo en la edición Registered.
TsgcGridRenderer, TsgcGridRenderOptions	Dibuja un fotograma de un **TsgcGridData** en un **TsgcUICanvas** y lo escribe como SVG o, con esegece.sgcGrid.Rendering, como PNG. Consulte Renderizado sin interfaz en .NET.
sgcGrid_IO	Importación y exportación de JSON y CSV, consulte Enlace con DataTable, JSON y CSV en .NET.
TsgcGridSource_DataSet	Enlaza un `System.Data.DataTable` a una cuadrícula, consulte Enlace con DataTable, JSON y CSV en .NET.
TsgcGridFeedSource y descendientes	Los orígenes de flujos en vivo de esegece.sgcGrid.sgcWebSockets, consulte Flujos en vivo en .NET. Solo en la edición Registered.
TsgcGridAIQuery	Preguntas en lenguaje natural de esegece.sgcGrid.sgcHTML, consulte Pregunte a su cuadrícula en .NET. Solo en la edición Registered.

Cuadrícula

Los mismos miembros en **TsgcWinFormsGrid**, **TsgcWpfGrid** y **TsgcAvaloniaGrid**.

Miembro	Tipo	Descripción
Columns	TsgcGridColumns	Las columnas.
Data	TsgcGridData	El almacén de filas, de solo lectura. Se puede escribir desde cualquier subproceso.
FormatRules	TsgcGridFormatRules	Las reglas de formato condicional.
Renderer	TsgcGridRenderer	El renderizador de la cuadrícula, de solo lectura.
HeaderHeight, RowHeight	int	Altura del encabezado y de una fila, 32 y 28 por defecto.
GridLinesH, GridLinesV, AlternateRows	bool	Líneas de cuadrícula horizontales y verticales y el tono de las filas alternas, true por defecto.
SelectionMode	TsgcGridSelectionMode	`gsmRow` (por defecto) o `gsmCell`.
Theme, AccentColor	TsgcUITheme, uint	Tema de la cuadrícula y un acento opcional `$AARRGGBB`, 0 para el acento del tema.
FontSize	double	Tamaño de fuente de las celdas y del encabezado, 13 por defecto.
MaxRows, TailMode	int, bool	Modo tail: las filas más antiguas se descartan a partir de **MaxRows**, y **TailMode** mantiene la última fila a la vista.
FilterText	string	Muestra solo las filas cuyo texto formateado lo contiene.

FlashChanges, FlashDuration, FlashMode	bool, int, TsgcGridFlashMode	Destello de los cambios, 800 ms por defecto, sobre el fondo o el texto.
FlashUpColor, FlashDownColor, FlashChangedColor	uint	Colores del destello, 0 para los colores del tema.
ReadOnly	bool	True por defecto. False activa la edición en el lugar de las columnas **Editable**. Solo en la edición Registered.
SelectedRow, SelectedCol, SelectedKey	int, int, string	La selección. La selección sigue a su fila por clave cuando se reordenan las filas.
TopRow, ScrollX, ScrollToRow	int, int, method	La posición de desplazamiento. **TopRow** cuenta líneas de visualización, que son las filas de la vista cuando no hay nada agrupado.
RowAt, ColAt	method	La fila de la vista y la columna bajo un punto.
StartEdit, CommitEdit, CancelEdit, EditorActive	method, bool	Edición en el lugar desde el código. Solo en la edición Registered.
SaveToSVG, SaveToPNG	method	Escribe la vista actual como imagen.
OnCellClick, OnCellDblClick	TsgcGridCellEvent	Se hizo clic en una celda, con la fila de la vista y el índice de la columna.
OnSelectionChange, OnSortChange	EventHandler	Cambió la selección o la ordenación.
OnColumnResized	TsgcGridColumnEvent	El usuario cambió el tamaño de una columna.
OnGetCellStyle	TsgcGridCellStyleEvent	Cambia los colores de una celda antes de dibujarla.
OnBeforeEdit, OnValidate, OnEdited, OnEditError	events	Edición en el lugar, consulte Edición en el lugar en .NET. Solo en la edición Registered.
ShowGroupPanel, GroupPanelHeight, GroupIndent, GroupsExpanded, GroupSummaries	bool, int, int, bool, bool	Opciones de visualización de la agrupación, consulte Agrupación y resúmenes de .NET.
ShowFooter, FrozenColumns	bool, int	El pie de resúmenes y las columnas que permanecen a la vista mientras las demás se desplazan.
Lines, FocusedLine	TsgcGridLineView, int	Las líneas de visualización y la línea con el foco.
GroupBy, ClearGrouping, ExpandAll, CollapseAll, ExpandGroup, CollapseGroup, GroupCount, GroupInfo, FooterSummary, FooterSummaryText	method	Agrupación y resúmenes desde código.
OnGroupingChange, OnGroupExpandChange, OnGetGroupText	eventos	La agrupación o el estado de un grupo cambió, y el título de una fila de grupo.

Tree, Detail	TsgcGrid-TreeOptions, TsgcGridDetai-lOptions	Filas de árbol y bandas de detalle, consulte Filas de árbol y maestro detalle de .NET. Solo en la edición Registered.
ExpandNode, CollapseNode, ExpandDetail, CollapseDetail, DetailExpanded, DetailData	method	Nodos de árbol y bandas de detalle desde código. Solo en la edición Registered.
OnTreeExpandChange, On-DetailExpandChange, OnDe-tailCellClick, OnGetDetailData	eventos	Eventos de árbol y de detalle. Solo en la edición Regis-tered.
SaveToXLSX	method	Escribe la vista como un libro de Excel, consulte Expor-tación a Excel de .NET. Solo en la edición Registered.
VertBarVisible, HorzBarVisi-ble, BorderStyle	bool, bool, TsgcGridBor-derStyle	WPF y Avalonia: las barras de desplazamiento dibuja-das y el borde. WinForms usa el **BorderStyle** de `System.Windows.Forms` y barras de desplazamiento nativas.

Consulte también

- Descripción general de .NET
- Diferencias entre .NET y Delphi
- Renderizado sin interfaz en .NET
- Enlace con DataTable, JSON y CSV en .NET
- Flujos en vivo en .NET
- Pregunte a su cuadrícula en .NET
- Agrupación y resúmenes de .NET
- Filas de árbol y maestro detalle de .NET
- Exportación a Excel de .NET
- TsgcGridData
- TsgcGridRenderer

Diferencias entre .NET y Delphi

El port de .NET conserva los nombres, los valores por defecto y las reglas de renderizado de Delphi; su salida SVG es idéntica byte a byte a la de Delphi. Las diferencias que siguen proceden de los frameworks de los hosts y de la biblioteca de clases de .NET.

Data y FormatRules en lugar de IsgcGridDataHost

En Delphi un origen encuentra los datos de una cuadrícula mediante la interfaz IsgcGridDataHost e IsgcUITable. El port de .NET no tiene esa interfaz. **TsgcWinFormsGrid**, **TsgcWpfGrid** y **TsgcAvaloniaGrid** exponen **Data** y **FormatRules** como propiedades públicas, y **TsgcGridSource_DataSet**, los orígenes de flujos y **TsgcGridAIQuery** las encuentran por reflexión: **Grid** acepta cualquier objeto con una propiedad pública **Data** de tipo **TsgcGridData**, y la consulta de IA también lee una propiedad pública **FormatRules** de tipo **TsgcGridFormatRules** para sus resaltados. Así, su propia clase puede ser el **Grid** de un origen, consulte el ejemplo en Pregunte a su cuadrícula en .NET. **GridData**, cuando está asignado, siempre prevalece sobre **Grid**.
En .NET no existe la notificación de liberación. Cuando **Grid** es un `IComponent`, como un control WinForms, el origen escucha su evento `Disposed` y entonces borra **Grid**. Para cualquier otro objeto, establezca **Grid** en null antes de que la cuadrícula desaparezca. **GridData** no se vigila.

Barras de desplazamiento

TsgcWinFormsGrid usa las barras de desplazamiento nativas de Windows, como la cuadrícula VCL. **TsgcWpfGrid** y **TsgcAvaloniaGrid** dibujan sus propias barras de desplazamiento, como la cuadrícula FireMonkey, con arrastre del control deslizante y paginación por la pista; **VertBarVisible** y **HorzBarVisible** las muestran u ocultan.

FontSize en WPF, Theme en Avalonia

La cuadrícula dibuja su propio texto, por lo que **TsgcWpfGrid** declara su propio **FontSize** con `new`, que oculta `Control.FontSize`. A través de una referencia de tipo `Control` es la propiedad de WPF, que la cuadrícula no lee. De la misma forma, todo control de Avalonia ya tiene una propiedad **Theme**, el `ControlTheme` de `StyledElement`: **TsgcAvaloniaGrid** declara su propio **Theme** de tipo **TsgcUITheme** con `new`. A través de una referencia del tipo de la cuadrícula es el tema sgc; a través de una referencia de tipo `Control` o `StyledElement` es el tema de control de Avalonia.
En WPF y Avalonia ambas propiedades admiten enlaces y estilos, igual que la selección y las demás opciones de visualización, consulte Enlace de datos en .NET.

Táctil

En Avalonia un contacto táctil desplaza la cuadrícula, el contenido sigue al dedo sin desplazamiento cinético, como hace el gesto de desplazamiento de FireMonkey, y un toque selecciona la celda. Allí un contacto táctil nunca arrastra un encabezado al panel de grupos; un toque en un chevron, un expansor o una banda de detalle actúa como un clic. WPF no tiene desplazamiento táctil: el tacto y el lápiz se promueven a entrada de ratón, de modo que un toque selecciona y las barras de desplazamiento dibujadas desplazan. WinForms usa el ratón y la rueda.

Tipos de datos

Los tipos de CLR se asignan a los tipos de datos de la cuadrícula como los tipos de campo de Delphi: `byte`, `short`, `int`, `long` y los demás tipos enteros se asignan a `gdtInteger`; `float`, `double` y `decimal` se asignan a `gdtFloat`, como los campos `ftBCD` y `ftFMTBcd` de Delphi; `DateTime`, `DateTimeOffset` y `TimeSpan` se asignan a `gdtDateTime`; `bool` se asigna a `gdtBoolean`; todo lo demás es `gdtText`. CLR no tiene un tipo currency, por lo que `gdtCurrency` solo procede de una columna que usted mismo añada.

DataTable en lugar de TDataSet

TsgcGridSource_DataSet enlaza un `System.Data.DataTable`, el equivalente en .NET de un `TDataSet` de Delphi. Un DataTable siempre está abierto y no tiene cursor, por lo que no hay registro actual ni marcador que restaurar después de una carga. Los eventos de **LiveUpdates** (`RowChanged`, `RowDeleted`, `TableCleared`) ocupan el lugar del data link de Delphi, y las ediciones sin terminar y `AcceptChanges` se ignoran. Consulte Enlace con DataTable, JSON y CSV en .NET.

Reglas de formato sin ID de elemento

Las reglas de formato de .NET no tienen ID de elemento de colección. **TsgcGridAIQuery.Reset** elimina las reglas que añadió por referencia en lugar de por ID; las columnas se siguen encontrando por su **ID**.

Edición de las filas de árbol, las bandas de detalle y la exportación a Excel

En Delphi, las filas de árbol, las bandas de detalle y **SaveToXLSX** se compilan bajo la definición `SGC_GRID_ADVANCED`. Los paquetes Community de .NET no contienen **TsgcGridTreeOptions**, **TsgcGridDetailOptions**, **TsgcGridDataStore**, **TsgcGridXLSXWriter** ni los miembros de la cuadrícula que los usan; la agrupación, los resúmenes, el pie y las columnas fijas están en ambas ediciones. Consulte Filas de árbol y maestro detalle de .NET y Exportación a Excel de .NET.

Eventos de agrupación, árbol y detalle

Los nuevos eventos de la cuadrícula son eventos de C#, como los demás eventos de las cuadrículas. Sus delegados, **TsgcGridGroupExpandEvent**, **TsgcGridRowExpandEvent** y **TsgcGridDetailCellEvent**, se declaran una sola vez en esegece.sgcGrid y los comparten los tres hosts. Los métodos sin argumentos, como **GroupCount**, siguen siendo métodos.

Origen del detalle

Detail.Source es un `object`: un **TsgcGridDataStore**, un **TsgcGridData** o cualquier objeto con una propiedad pública **Data** de tipo **TsgcGridData**, que se encuentra como los orígenes encuentran una cuadrícula. Sin una notificación de liberación, un Source que es un `IComponent` se borra cuando se dispara su evento `Disposed`.

Archivos de Excel

El libro se comprime con `System.IO.Compression.DeflateStream`. Sus partes XML son los mismos bytes que la salida de Delphi para los mismos datos; solo difieren los bytes comprimidos y, por tanto, el tamaño del archivo.

Serialización en el diseñador de las nuevas opciones

No hay DFM. En WinForms las siete opciones de agrupación tienen valores predeterminados y se serializan solo cuando difieren, y **Tree** y **Detail** se serializan como contenido. En WPF y Avalonia las opciones son propiedades de dependencia y propiedades con estilo.

Sin verbos de diseñador

Las cuadrículas de Delphi añaden verbos de tiempo de diseño al IDE. Las cuadrículas de .NET no tienen ninguno: añada las **Columns** y las **FormatRules** en código, como hacen los ejemplos de inicio rápido.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Inicio rápido WPF
- Inicio rápido Avalonia
- Enlace con DataTable, JSON y CSV en .NET
- Pregunte a su cuadrícula en .NET
- Filas de árbol y maestro detalle de .NET
- Exportación a Excel de .NET
- IsgcGridDataHost e IsgcUITable

Enlace de datos en .NET

Este tema trata solo **TsgcWpfGrid** y **TsgcAvaloniaGrid**. Sus propiedades escalares visibles para el usuario son propiedades de dependencia de WPF y propiedades con estilo de Avalonia, por lo que admiten enlaces, estilos y setters XAML, y un modelo de vista puede controlar la cuadrícula sin código subyacente. **TsgcWinFormsGrid** no cambia: sus propiedades son propiedades CLR simples, como antes.

Propiedades enlazables

Las propiedades siguientes admiten un enlace en ambas cuadrículas. Las marcadas como TwoWay se enlazan TwoWay de forma predeterminada, de modo que un enlace sin `Mode` también lleva de vuelta al modelo de vista los cambios de la cuadrícula. Todas las demás se enlazan OneWay de forma predeterminada.

Propiedad	Tipo	Modo predeterminado
SelectedRow	int	TwoWay
SelectedCol	int	TwoWay
Theme	TsgcUITheme	OneWay
FontSize	double	OneWay
BorderStyle	TsgcGridBorderStyle	OneWay
ReadOnly	bool	OneWay
TailMode	bool	OneWay
MaxRows	int	OneWay
FilterText	string	OneWay
HeaderHeight	int	OneWay
RowHeight	int	OneWay
GridLinesH	bool	OneWay
GridLinesV	bool	OneWay
AlternateRows	bool	OneWay
SelectionMode	TsgcGridSelectionMode	OneWay
AccentColor	uint	OneWay
FlashChanges	bool	OneWay
FlashDuration	int	OneWay
FlashUpColor	uint	OneWay
FlashDownColor	uint	OneWay
FlashChangedColor	uint	OneWay

FlashMode	TsgcGridFlashMode	OneWay

SelectedRow y **SelectedCol** siguen un clic, una tecla o el código. En Avalonia son propiedades directas. En WPF **FontSize** se declara con `new` y oculta `Control.FontSize`, y en Avalonia **Theme** oculta el tema de control, consulte Diferencias de .NET respecto a Delphi. Enlácelas en el propio elemento de la cuadrícula, como hacen los ejemplos.

Comportamiento

- El modelo de la cuadrícula conserva su validación. Cuando rechaza o corrige un valor, por ejemplo un **FlashDuration** negativo o un **SelectedRow** posterior a la última fila, la cuadrícula escribe de vuelta en la propiedad el valor que conservó. En WPF, cuando el valor procedía de un origen TwoWay, el valor conservado también vuelve a ese origen un tick del dispatcher más tarde, cuando el enlace ha terminado su transferencia.
- Cuando el código cambia el modelo directamente mediante **Renderer.Options**, el nuevo valor se copia de vuelta en la propiedad enlazable solo en el subproceso de la interfaz, de modo que un modelo de vista enlazado nunca se actualiza desde otro subproceso.
- **OnSelectionChange** y los demás eventos se generan ante cualquier cambio, ya provenga de un enlace, un estilo, el código, el ratón o el teclado.
- La selección sigue a su clave de fila cuando **FilterText** mueve la fila, y el **SelectedRow** enlazado también la sigue.

Propiedades que siguen siendo CLR

- **Data**, **Columns** y **FormatRules** siguen siendo propiedades CLR, porque **TsgcGridSource_DataSet**, los orígenes de flujo y **TsgcGridAIQuery** las encuentran por reflexión. Rellénelas desde el código o mediante un origen, consulte Enlace con DataTable, JSON y CSV en .NET y Flujos en vivo en .NET.
- **TopRow** y **ScrollX** siguen siendo propiedades CLR, porque son estado de la vista que cambia en cada paso de desplazamiento.
- Las propiedades de estado de solo lectura, como **SelectedKey**, **VertBarVisible**, **HorzBarVisible** y **Renderer**, siguen siendo propiedades CLR, porque no hay nada que enlazar a ellas.

Modelo de vista

Ambos ejemplos usan el mismo modelo de vista. **SelectedRow** está enlazado TwoWay, **FilterText** se comparte con un cuadro de texto, y **Theme** se calcula a partir de una casilla de verificación.
MainViewModel.cs:

Ejemplo WPF

MainWindow.xaml:
MainWindow.xaml.cs:

Ejemplo Avalonia

La ventana de Avalonia usa enlaces compilados, por lo que establece `x:DataType` con el modelo de vista.
MainWindow.axaml:
MainWindow.axaml.cs:
Establecer `vm.SelectedRow = 3` selecciona la fila AMZN, y un clic en otra fila escribe su índice de vuelta en `vm.SelectedRow`. Escribir en el cuadro de texto filtra la cuadrícula, y la casilla de verificación cambia el tema.

Consulte también

- Inicio rápido WPF
- Inicio rápido Avalonia

- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Enlace con DataTable, JSON y CSV en .NET
- Flujos en vivo en .NET

Renderizado sin interfaz en .NET

El renderizador no necesita ningún control ni ventana, por lo que un servicio, una aplicación web o un servidor Linux pueden escribir la imagen de una cuadrícula. **TsgcGridRenderer.SaveToSVG** escribe SVG desde el paquete core. **SaveToPNG**, un método de extensión de esegece.sgcGrid.Rendering, escribe PNG mediante SkiaSharp.

Ejemplo

Este programa de consola construye el modelo, lo ordena, añade una regla de formato y escribe el mismo fotograma como SVG y como PNG al doble de tamaño:
El programa escribe:

```
North America: 318,950.00
Europe: 245,300.50
Asia Pacific: 162,420.75
Latin America: 71,880.20
regions.svg 6846 bytes, regions.png 49632 bytes
```

Notas

- El ancho y el alto están en píxeles independientes del dispositivo; el último argumento de **SaveToSVG** y **SaveToPNG** es la escala, así que `520 x 170` con escala 2 escribe una imagen de 1040 x 340 píxeles.
- `sgcGrid_Render.sgcGridRenderState()` devuelve un estado sin selección y sin desplazamiento. Establezca sus campos para renderizar un fotograma desplazado o con selección.
- Las opciones de renderizado (**Theme**, **FontSize**, **RowHeight**, los ajustes de destello y **FormatRules**) son las que exponen los controles de cuadrícula, y un control de cuadrícula escribe su propia vista con **SaveToSVG** y **SaveToPNG**.
- El texto SVG es idéntico byte a byte al que escribe el renderizador de Delphi para los mismos datos. La salida PNG usa el suavizado de Skia, por lo que puede diferir de la trama de Delphi en píxeles sueltos.
- **UseInvariantFormat** hace que **CellText** sea independiente de la cultura de la máquina.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Rendimiento en .NET
- Renderizado sin interfaz
- TsgcGridRenderer

Enlace con DataTable, JSON y CSV en .NET

TsgcGridSource_DataSet enlaza un `System.Data.DataTable` a una cuadrícula, y la clase **sgcGrid_IO** carga y guarda las filas como JSON y CSV. Ambos están en el paquete core y funcionan con todos los hosts y sin interfaz.

Enlace con DataTable

- Establezca **Grid** en el control de cuadrícula, o **GridData** en su **Data**.
- Con **AutoColumns** true (el valor por defecto) y una cuadrícula sin columnas, las columnas se crean a partir del DataTable: el nombre del campo es `ColumnName`, el título es `Caption`, una columna con `ColumnMapping Hidden` queda oculta, y el tipo de datos sigue el tipo CLR (`decimal` se asigna a `gdtFloat`).
- **KeyField** indica la columna cuyo valor es la clave de la fila. Con una clave, una recarga actualiza las filas en el lugar y conserva su destello de cambio, añade las claves nuevas y elimina las claves que han desaparecido. Sin clave, cada carga sustituye las filas.
- Con **LiveUpdates** true (el valor por defecto), un cambio confirmado, una eliminación o un vaciado del DataTable llega a la cuadrícula al instante. **Load** carga en cualquier momento y **Refresh** recarga.

El programa escribe:

```
sku "sku" gdtText
name "Product" gdtText
stock "Stock" gdtInteger
price "Price" gdtFloat
3 rows
A-200: Mouse, stock 95
B-300: Monitor, stock 8
B-400: Webcam, stock 30
```

Asignar null a **DataSet** desenlaza la tabla y conserva las filas. Liberar el DataTable hace lo mismo.

JSON y CSV

GridLoadFromJSON lee un array de objetos; un nombre de campo con puntos como `weather.temp` lee un objeto anidado, y **GridSaveToJSON** lo vuelve a escribir como tal. **GridLoadFromCSV** y **GridSaveToCSV** usan las comillas de la RFC 4180, cualquier delimitador y una fila de encabezado. El argumento del campo clave indica la clave de la fila y, con `aAutoColumns`, las columnas se crean a partir del primer objeto o de la fila de encabezado, con los tipos deducidos de los valores. Las sobrecargas de cadena, flujo y archivo leen UTF-8 y omiten un BOM, y escriben UTF-8 sin él.
El programa escribe:

```
2 rows from JSON, key of row 0: 1
id,city,weather.temp,weather.sky
1,Barcelona,21.5,Sunny
2,Oslo,8,"Rain, light"
4 columns, 2 rows from CSV, Temp is gdtFloat
[{"id":1,"city":"Barcelona","weather":{"temp":21.5,"sky":"Sunny"}},{"id":2,"city":"Oslo","weather":{"temp":8,"sky":"Rain, l
```

Los números CSV usan la cultura invariante salvo que `aInvariant` sea false; en ese caso se usan los **FormatSettings** de los datos. Las filas se escriben en el orden de la vista, y `aVisibleOnly` escribe solo las columnas visibles.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias entre .NET y Delphi
- TsgcGridSource_DataSet
- Enlace con DataSet
- JSON y CSV

Edición en el lugar en .NET

Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.
Las tres cuadrículas editan una celda en el lugar cuando **ReadOnly** es false y la columna es **Editable**. Sobre la celda se abre un cuadro de texto, o un cuadro combinado cuando la columna tiene una **PickList**. Una celda booleana se alterna con la barra espaciadora o con un clic. La edición comienza con F2, Intro, un carácter escrito, un doble clic o **StartEdit**, se confirma con Intro, Tab o cuando el editor pierde el foco, y se cancela con Escape o **CancelEdit**.

Eventos

Evento	Descripción
OnBeforeEdit	Antes de que se abra el editor. Establezca `aAllow` en false para vetar la edición.
OnValidate	Antes de escribir el valor, ya convertido al **DataType** de la columna. Cambie `aValue`, o establezca `aAccept` en false para rechazarlo; el editor permanece entonces abierto.
OnEdited	Después de escribir el valor con **Data.UpdateCell**, con el valor anterior y el nuevo.
OnEditError	El texto no se puede convertir al **DataType** de la columna. Cambie `aMessage` para establecer el mensaje.

El argumento de columna de los eventos de edición es el índice en **Columns**, no la posición visible.

Ejemplo

Un flujo en vivo nunca sobrescribe un editor abierto: el editor sigue a su fila por clave cuando se reordenan las filas y se cierra cuando se elimina la fila. En WinForms los editores son los controles hijos **TsgcWinFormsGridEdit** y **TsgcWinFormsGridCombo**; en WPF y Avalonia son un `TextBox` y un `ComboBox` alojados. Cuando un mismo proyecto compile ambas ediciones, proteja el código de edición con `#if !SGC_COMMUNITY`.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Inicio rápido WinForms
- Edición en el lugar
- Selección y teclado
- Ediciones

Flujos en vivo en .NET

Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.
El paquete esegece.sgcGrid.sgcWebSockets alimenta una cuadrícula desde conexiones de sgcWebSockets .NET. **TsgcGridFeedSource** es la clase base; **TsgcGridFeedSource_WebSocket** lee un **TsgcWebSocketClient**, **TsgcGridFeedSource_MQTT** lee las publicaciones de un cliente MQTT y **TsgcGridFeedSource_SSE** lee server-sent events. Cada mensaje se lee como JSON y se asigna a filas de la cuadrícula.

Asignación

Propiedad	Descripción
Grid, GridData	El destino: un control de cuadrícula, encontrado a través de su propiedad pública **Data**, o un **TsgcGridData**. **GridData** prevalece.
RowsPath	Ruta de las filas en el mensaje: un array de objetos de fila o un único objeto de fila. Vacía cuando el propio mensaje es la fila o el array.
KeyPath	Ruta de la clave de la fila dentro de un objeto de fila, por ejemplo `symbol` o `id.value`. Una fila sin clave se omite y se notifica mediante **OnError**.
Fields	Un `FieldName=json.path` por línea, por ejemplo `price=quote.last` o `bid=book.bid[0]`. Una línea sin `=` lee la ruta del mismo nombre. Cuando **Fields** está vacío, cada columna se lee de la ruta de su **FieldName**. Solo se escriben las rutas presentes en una fila, de modo que un mensaje puede llevar una actualización parcial.
DeletePath	Una ruta booleana en el objeto de fila: true elimina la fila de la clave.
MaxRowsPerMessage	Limita las filas aplicadas desde un mensaje, 0 para no limitar. Las filas sobrantes se notifican mediante **OnError**.
Active	False ignora todos los mensajes.
OnRow, OnError	Una fila está a punto de aplicarse (establezca `aAccept` en false para omitirla), o un mensaje no se pudo aplicar.

Los números JSON se escriben como double, los booleanos como bool, las cadenas como string y null como el null de la cuadrícula.

Orígenes

Clase	Propiedad del cliente	Lee
TsgcGridFeedSource_WebSocket	**Client** (`TsgcWebSocketClient`)	Todos los mensajes de texto.
TsgcGridFeedSource_MQTT	**Client** (`TsgcWSProtocol_MQTT_Client`)	Las publicaciones cuyo tema coincide con **Topic**, con los comodines `+` y `#`.
TsgcGridFeedSource_SSE	**Client** (`TsgcHTTP1Client`) o **SSEClient** (`TsgcSSEClient`)	Los datos de cada server-sent event. **SSEClient** es una incorporación de .NET.

Los orígenes se suscriben a los eventos del cliente junto a los manejadores de la aplicación y nunca los sustituyen.

Ejemplo

Este programa de consola inicia un **TsgcWebSocketServer** en el mismo proceso, en la interfaz de bucle invertido, y alimenta un modelo de cuadrícula con tres mensajes: un lote de filas, una actualización parcial con una eliminación y un único objeto de fila.
El programa escribe:

```
AAPL 228.05 x 300
NVDA 121.40 x 900
JPM 212.80 x 50
```

Subprocesos

Todas las filas de un mensaje se aplican dentro de un único **BeginUpdate** y **EndUpdate**, en el subproceso en el que llega el mensaje. **TsgcGridData** es seguro para subprocesos y las cuadrículas derivan su repintado al subproceso de la interfaz, por lo que no hace falta ninguna sincronización. **OnRow** y **OnError** también se ejecutan en ese subproceso: cambie al subproceso de la interfaz antes de tocar allí un control.

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Demos de .NET
- TsgcGridFeedSource
- TsgcGridFeedSource_WebSocket
- TsgcGridFeedSource_MQTT
- TsgcGridFeedSource_SSE
- Flujos en vivo

Pregunte a su cuadrícula en .NET

Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.
TsgcGridAIQuery, del paquete esegece.sgcGrid.sgcHTML, convierte una pregunta en lenguaje natural en filtros, ordenaciones y resaltados de una cuadrícula. La pregunta la responde el **TsgcHTMLAIAssistant** de esegece.sgcHTML: un modelo de lenguaje cuando el asistente tiene un proveedor, y su analizador basado en reglas cuando **Assistant** es null o no tiene proveedor. El analizador basado en reglas no necesita red y entiende preguntas como `price greater than 100`, `sort by volume descending` y `highlight change less than 0`.

Cómo se aplica la respuesta

- La respuesta se comprueba con las columnas de la cuadrícula (**FieldName**, **Caption** y **DataType**) antes de aplicar nada.
- Los filtros se combinan con AND mediante **Data.OnFilterRow**. Un manejador que usted haya asignado se conserva y se llama primero, y una fila que rechaza sigue rechazada.
- Las ordenaciones sustituyen el **SortOrder** y el **SortIndex** de las columnas.
- Los resaltados se añaden a las **FormatRules** de la cuadrícula con **ApplyToRow**, con los colores del estilo del asistente.
- La cuadrícula no admite agrupar ni limitar filas; se notifican en **LastExplanation**.
- Cada **Ask** deshace primero el anterior. **Reset** restaura el manejador de filtro y la ordenación de columnas anteriores y elimina solo las reglas que añadió la consulta.

Ask es síncrono y debe llamarse desde el subproceso de la interfaz, ya que cambia las columnas y las reglas de formato. Con un modelo de lenguaje se bloquea mientras el proveedor responde. **OnApplied** y **OnError** notifican el resultado.

Ejemplo

El **Grid** de la consulta es cualquier objeto con una propiedad pública **Data**; para los resaltados también necesita una propiedad pública **FormatRules**. Los controles de cuadrícula tienen ambas. Este programa de consola usa una pequeña clase propia y ningún asistente, por lo que responde el analizador basado en reglas:
El programa escribe:

```
Filter: Price > 120
Sort: Volume descending
Highlight (warning): Change < 0
NVDA 98000
AAPL 51000
MSFT 22000
JPM 9000
1 highlight rule
5 rows after Reset
```

Para usar un modelo de lenguaje, asigne a **Assistant** un **TsgcHTMLAIAssistant** con un proveedor configurado.

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Diferencias entre .NET y Delphi
- Demos de .NET
- TsgcGridAIQuery
- Pregunte a su cuadrícula

Rendimiento en .NET

La cuadrícula dibuja solo las filas y columnas visibles y se repinta como máximo **MaxFPS** veces por segundo, por rápido que cambien los datos. El host WinForms se mide con una puerta de rendimiento en la suite de pruebas, un port de la prueba de rendimiento de Delphi al que se han añadido la ruta de dibujo del host y un flujo en vivo.

La escena

- Un área de cliente de `1920 x 1080` de un **TsgcWinFormsGrid**, 12 columnas de 160 píxeles independientes del dispositivo (texto, entero, float, currency, hora y booleano) y una altura de fila de 23.
- 100.000 filas con clave.
- Un subproceso de trabajo envía 1000 actualizaciones por segundo, la mitad de ellas a filas cercanas a la parte superior de la vista.

Tiempo de fotograma

El tiempo de fotograma es el dibujo completo del control, renderizado y blit, durante 60 fotogramas mientras el flujo está activo. El presupuesto es de 16 ms por fotograma en un proceso de 64 bits, un fotograma a 60 fps, y de 25 ms en un proceso de 32 bits.

Runtime	Tiempo de fotograma
.NET 8, antes de las máscaras de cobertura de glifos de sgcUICore	44 ms
.NET 8	11,4 a 13,1 ms
.NET Framework 4.7.2	7,3 a 7,7 ms

La primera fila es la medición que condujo a las máscaras de cobertura de glifos de sgcUICore: cada texto de celda rellenaba todo su contorno mediante Skia.

Ritmo

La segunda fase de la puerta mide la frecuencia de repintado sostenida con **MaxFPS** en 60 sobre un bucle de mensajes real durante 3 segundos, y comprueba que el flujo mantiene al menos el 90 por ciento de su frecuencia y que el repintado nunca supera **MaxFPS**. Los rangos siguientes proceden de tres ejecuciones de la puerta.

Runtime	Frecuencia de repintado
.NET 8	59,5 a 59,7 fps
.NET Framework 4.7.2	59,8 a 60 fps

En .NET 8 el percentil 95 del intervalo entre dos fotogramas es de 21,4 a 22,2 ms.

Ejecutar la puerta

La puerta es sensible al tiempo y necesita una máquina por lo demás inactiva, por lo que solo se ejecuta bajo petición: establezca la variable de entorno `SGC_GRID_RUN_PERF=1` y ejecute la categoría `Perf` de las pruebas de WinForms.
Las cifras se añaden a `sgcGridQA_perf.txt`, junto al ensamblado de pruebas. La puerta también escribe, como referencia, los tiempos de **SaveToPNG** y de **Render** sin interfaz del mismo fotograma. El **Render** sin interfaz de este fotograma tarda 5,0 ms.

Consulte también

- Descripción general de .NET
- Inicio rápido WinForms
- Renderizado sin interfaz en .NET
- Rendimiento
- Actualizaciones en vivo y subprocesos

Agrupación y resúmenes de .NET

Las filas se pueden agrupar por una o varias columnas. Cada grupo tiene una fila de grupo con un chevron para expandir, el texto del miembro, el número de filas y los resúmenes de las columnas; un pie muestra los mismos resúmenes sobre todas las filas, y las columnas fijas permanecen a la vista mientras las demás se desplazan horizontalmente. La agrupación, los resúmenes, el pie y las columnas fijas forman parte de ambas ediciones, en **TsgcWinFormsGrid**, **TsgcWpfGrid** y **TsgcAvaloniaGrid**.
La agrupación es estado de la vista. **TsgcGridData** conserva sus filas planas y su ordenación, e **IsgcUITable** sigue informando de las filas de la vista sin agrupar, de modo que un gráfico de sgcCharts enlazado a la cuadrícula no ve ningún cambio. La cuadrícula distribuye las filas mediante un **TsgcGridLineView**, disponible en su propiedad **Lines**.

Propiedades de columna

Propiedad	Valor predeterminado	Descripción
GroupIndex	-1	Un valor de 0 o más agrupa las filas por la columna en ese nivel. Los niveles se ordenan por **GroupIndex** y después por el índice de columna. **GroupBy**, **ClearGrouping** y el panel de grupos lo establecen por usted.
SummaryType	`gsuNone`	`gsuSum`, `gsuCount`, `gsuAverage`, `gsuMin` o `gsuMax`. El resumen se muestra en cada fila de grupo y en el pie.
SummaryFormat	vacío	El formato del resumen, como en **DisplayFormat**: por ejemplo `#,##0.00`, con texto literal entre comillas dobles, `"Total: "#,##0.00`. Vacío usa **DisplayFormat**; un recuento usa `0`.

Cambiar estas propiedades solo dispara el **OnChange** de las columnas y nunca recarga los datos. Los valores del resumen son `null` cuando no hay valor, un `int` para un recuento, un `double` para una suma, un promedio, un mínimo o un máximo, y un `DateTime` para el mínimo o el máximo de una columna de fecha.

Miembros de la cuadrícula

Los mismos miembros existen en **TsgcWinFormsGrid**, **TsgcWpfGrid** y **TsgcAvaloniaGrid**. En WPF y Avalonia las siete opciones son propiedades de dependencia y propiedades con estilo, por lo que se pueden establecer en XAML, enlazar y aplicar estilos.

Miembro	Tipo	Descripción
ShowGroupPanel, GroupPanelHeight	bool, int	El panel sobre el encabezado: arrastre allí el encabezado de una columna para agrupar por ella. Oculto de forma predeterminada, con 36 píxeles de alto.
GroupIndent	int	Sangría de cada nivel de grupo, 20 de forma predeterminada.
GroupsExpanded	bool	El estado de los grupos nuevos, true de forma predeterminada.
GroupSummaries	bool	Muestra los resúmenes en las filas de grupo, true de forma predeterminada.
ShowFooter	bool	Muestra el pie con los resúmenes sobre todas las filas, false de forma predeterminada.

FrozenColumns	int	Las columnas visibles iniciales que permanecen a la vista mientras las demás se desplazan, 0 de forma predeterminada.
GroupBy, ClearGrouping	method	`GroupBy("region", "category")` agrupa por estos campos en este orden y borra cualquier otro **GroupIndex**; **ClearGrouping** vuelve a mostrar las filas planas.
ExpandAll, CollapseAll, ExpandGroup, CollapseGroup	method	Expanden o contraen todos los grupos, o un grupo por su índice.
GroupCount, GroupInfo	method	El número de grupos y el **TsgcGridGroupInfo** de un grupo.
FooterSummary, FooterSummaryText	method	El valor del pie de una columna (null cuando no hay ninguno) y su texto tal como lo muestra el pie.
Lines	TsgcGridLineView	La vista de líneas de la cuadrícula: filas de grupo, filas de datos y, en la edición Registered, filas de árbol y bandas de detalle.
FocusedLine	int	La línea con el foco. En una fila de grupo **SelectedRow** es -1 y **FocusedLine** indica la línea.
OnGroupingChange	EventHandler	Las columnas agrupadas o su orden han cambiado, desde código o desde el panel de grupos.
OnGroupExpandChange	TsgcGridGroupExpandEvent	Un grupo se expandió o se contrajo. `aGroup` es -1 después de **ExpandAll** y **CollapseAll**.
OnGetGroupText	TsgcGridGetGroupTextEvent	Cambia el título de una fila de grupo antes de dibujarla.

Los índices de fila siguen siendo filas de la vista: **SelectedRow**, **OnCellClick** y **Data** usan las mismas filas que sin agrupación. **TopRow** cuenta líneas de visualización, que son las filas de la vista cuando no hay nada agrupado.

Información del grupo

Field	Descripción
Level	0 para el nivel más externo.
Parent	El grupo padre, -1 en el nivel 0.
Column	El índice de **Columns** de la columna agrupada.
RowCount	Las filas de la vista del grupo, incluidos sus grupos hijos.
FirstRow	La fila de la vista de la primera fila del grupo.
Text, Value, Blank	El texto del miembro, su valor tipado, y true para el miembro en blanco, que se muestra como `(blank)`.
Expanded	El grupo está expandido.

Line	La línea de la fila de grupo, -1 dentro de un grupo contraído.
Path	Los textos de los miembros desde el nivel 0.

Ratón y teclado

- Arrastre el encabezado de una columna al panel de grupos para agrupar por ella; una marca indica dónde se suelta. Arrastre un chip a lo largo del panel para cambiar el orden, arrástrelo fuera para quitarlo, haga clic en él para invertir la ordenación del grupo. Escape cancela un arrastre.
- Haga clic en el chevron de una fila de grupo, o haga doble clic en la fila, para expandirla o contraerla.
- En una fila de grupo, Right y Left expanden y contraen, + y la tecla menos hacen lo mismo, * expande todos los grupos y Enter alterna el grupo. Ctrl+C en una fila de grupo copia su título.
- En Avalonia un contacto táctil sigue desplazando la cuadrícula; un toque en un chevron actúa como un clic.

Ejemplo

Un formulario WinForms que agrupa pedidos por región y categoría, con una suma y un promedio en cada fila de grupo y en el pie, dos columnas fijas y un título personalizado para los grupos de región:
Los datos se pueden escribir desde cualquier subproceso mientras están agrupados: la cuadrícula resincroniza los grupos y los resúmenes en el subproceso de la interfaz y dosifica el trabajo, de modo que la interfaz sigue respondiendo.

Agrupación sin control

TsgcGridLineView funciona solo con un **TsgcGridData**, por ejemplo en un servicio o en un programa de consola. **GroupSummary** devuelve el valor del resumen de un grupo y una columna, y **SetGroupExpanded** expande o contrae un grupo:
Salida:

```
2 levels, 6 groups, 11 lines
North: 3 rows, total 142.5
  Books: 2 rows, total 82.5
  Games: 1 rows, total 60
South: 2 rows, total 140.9
  Books: 1 rows, total 41
  Games: 1 rows, total 99.9
6 lines with North collapsed
```

Unos datos y una vista de líneas son también lo que escribe **TsgcGridXLSXWriter**, consulte Exportación a Excel de .NET.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Filas de árbol y maestro detalle de .NET
- Exportación a Excel de .NET
- Demos de .NET
- TsgcGridColumn
- Agrupación
- Resúmenes
- Columnas fijas
- TsgcGridLineView

Filas de árbol y maestro detalle de .NET

Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen.
TsgcWinFormsGrid, **TsgcWpfGrid** y **TsgcAvaloniaGrid** muestran las filas como un árbol, construido a partir de un campo de clave y un campo padre, y abren una banda de detalle bajo una fila maestra con sus filas hijas de un segundo origen. El modo árbol y la agrupación no están activos a la vez: en modo árbol la agrupación se ignora, el **GroupIndex** de las columnas se conserva y el panel de grupos se oculta.

Filas de árbol

La propiedad **Tree** es un **TsgcGridTreeOptions**. El árbol está activo cuando **KeyField** y **ParentField** están ambos establecidos.

Propiedad	Valor predeterminado	Descripción
KeyField	vacío	La columna con la clave de la fila.
ParentField	vacío	La columna con la clave de la fila padre. Las filas con una clave de padre vacía, desconocida o propia son raíces.
TreeField	vacío	La columna que muestra la sangría y el botón de expandir. Vacío usa la primera columna visible.
Indent	16	Sangría de cada nivel.
Expanded	false	El estado de los nodos nuevos.

Los campos deben ser columnas de la cuadrícula; pueden estar ocultas. Las claves se comparan como texto: los enteros y los números de coma flotante sin fracción usan su texto entero, de modo que una clave `int` y un padre `double` coinciden. Cuando varias filas comparten una clave, la primera fila es la propietaria de los hijos, y un ciclo se rompe en el primer nodo repetido. Los hermanos conservan el orden de la vista, por lo que ordenar ordena los hermanos, y con un filtro una fila visible cuelga de su antecesor visible más cercano. El estado expandido se conserva por clave, por lo que sobrevive a una recarga.

Miembro	Descripción
ExpandNode, CollapseNode	Expanden o contraen el nodo de una fila de la vista.
ExpandAll, CollapseAll	En modo árbol expanden o contraen todos los nodos.
OnTreeExpandChange	Un nodo se expandió o se contrajo; `aRow` es -1 después de **ExpandAll** y **CollapseAll**.

Haga clic en el botón de un nodo para expandirlo o contraerlo. En un nodo, Right lo expande o pasa a su primer hijo, y Left lo contrae o pasa a su padre.

Maestro detalle

La propiedad **Detail** es un **TsgcGridDetailOptions**. Una columna de expansión abre una banda bajo una fila maestra; la banda es una cuadrícula anidada con su propio encabezado, posición de desplazamiento y selección. Los detalles se anidan un solo nivel y son de solo lectura.

Propiedad	Valor predeterminado	Descripción
Active	false	Muestra la columna de expansión.
Source	null	Las filas de detalle: un **TsgcGridDataStore**, un **TsgcGridData**, o cualquier objeto con una propiedad pública **Data** de tipo **TsgcGridData**, como otra cuadrícula.
MasterKeyField	vacío	La columna maestra cuyo valor se busca en las filas de detalle.
DetailKeyField	vacío	La columna del origen que contiene la clave maestra.
Lines	8	Altura de una banda en filas de la cuadrícula, incluido el encabezado anidado.
Columns	vacío	Las columnas de las bandas. Vacío usa una copia de las columnas del origen.

Miembro	Descripción
ExpandDetail, CollapseDetail, DetailExpanded	Abren, cierran y comprueban la banda de una fila de la vista.
DetailData	El **TsgcGridData** de una banda abierta, null cuando la banda está cerrada.
OnDetailExpandChange	Una banda se abrió o se cerró.
OnDetailCellClick	Se hizo clic en una celda de una banda: fila maestra, fila de detalle y columna de detalle.
OnGetDetailData	Se dispara antes de rellenar una banda. Escriba las filas en `aData` y establezca `aHandled` en true para rellenar la banda desde código; de lo contrario la banda se rellena desde **Source**.

El estado expandido se conserva por clave maestra. Cuando cambian las filas del origen, cada banda abierta se vuelve a rellenar en el subproceso de la interfaz, en su sitio. Al hacer clic en un encabezado de una banda se ordena, la rueda del ratón sobre una banda la desplaza, y Down pasa de largo una banda. La imagen de la banda se dibuja mediante esegece.sgcGrid.Rendering, al que hace referencia cada paquete de cuadrícula.
Cuando **Source** es un componente, al liberarlo se borra **Source**. Para cualquier otro objeto, establezca **Source** en null antes de que el objeto desaparezca.

TsgcGridDataStore

TsgcGridDataStore es un componente no visual que contiene filas de detalle: **Columns**, **Data** e **IsgcUITable**. Rellénelo desde código, o establézcalo como **Grid** de un **TsgcGridSource_DataSet** para cargar en él un `DataTable`. Establezca **Active** en true después de **Grid** y **DataSet**, y **LiveUpdates** mantiene el almacén sincronizado, consulte Enlace con DataTable, JSON y CSV en .NET.

Ejemplo

Un organigrama como árbol, y clientes con una banda de sus pedidos desde un **TsgcGridDataStore**:
La segunda clase rellena las bandas solo desde código: sin **Source** las bandas usan las **Columns** de las opciones, y cada fila procede de **OnGetDetailData**.

Sin un control

TsgcGridLineView admite las mismas opciones: asigne un **TsgcGridTreeOptions** a **TreeOptions** y un **TsgcGrid-DetailOptions** a **DetailOptions**; **SetNodeExpanded** y **SetDetailExpanded** cambian el estado. La exportación a Excel lo usa para escribir un árbol sin un control, consulte Exportación a Excel de .NET.

Consulte también

- Clases de .NET
- Agrupación y resúmenes de .NET
- Exportación a Excel de .NET
- Enlace con DataTable, JSON y CSV en .NET
- Demos de .NET
- TsgcGridSource_DataSet
- Filas de árbol
- Maestro detalle
- TsgcGridDataStore

Exportación a Excel de .NET

Esta característica forma parte de la edición Registered. Los paquetes Community no la incluyen. **TsgcWinFormsGrid**, **TsgcWpfGrid** y **TsgcAvaloniaGrid** guardan su vista como un libro de Excel (.xlsx) sin instalar Excel y sin ninguna biblioteca externa. El archivo lo escribe **TsgcGridXLSXWriter** del paquete esegece.sgcGrid, que también funciona sin un control.

Qué se escribe

- Las columnas visibles en su orden, los títulos en una fila de encabezado en negrita, y los anchos de columna.
- Números, fechas y booleanos como celdas tipadas, con el formato numérico de Excel del **DisplayFormat** de la columna; el texto como cadenas compartidas.
- Los grupos como filas de grupo con su título y fórmulas `SUBTOTAL` para los resúmenes de las columnas, sobre sus filas y en niveles de esquema de Excel. Los grupos contraídos se escriben como filas ocultas, de modo que el libro se abre como lo muestra la cuadrícula.
- Cuando una columna tiene un resumen, una fila de pie bajo los datos con `SUBTOTAL` sobre todo el rango de datos.
- Las filas de árbol con sangría en la columna de árbol y en niveles de esquema; las filas de los nodos contraídos son filas ocultas.
- Las columnas fijas y la fila de encabezado como panel inmovilizado.

Las bandas de detalle no se escriben. La ordenación y el filtrado se aplican: el libro contiene las filas de la vista.

SaveToXLSX

`SaveToXLSX(Stream aStream, string aSheetName = "Sheet1")` y `SaveToXLSX(string aFileName, string aSheetName = "Sheet1")` escriben la vista con los anchos de columna en pantalla y **FrozenColumns**. Llámelos en el subproceso de la interfaz.

TsgcGridXLSXWriter

Miembro	Descripción
SheetName	Nombre de la hoja de cálculo, `Sheet1` de forma predeterminada. Los caracteres que Excel rechaza se convierten en `_` y el nombre se corta a 31 caracteres.
FrozenColumns	Las columnas visibles iniciales que permanecen a la vista, 0 de forma predeterminada.
Summaries	Escribe las filas `SUBTOTAL` de grupo y de pie, true de forma predeterminada.
FileTime	Fecha y hora de las entradas del archivo, la hora de creación de forma predeterminada.
SetColumnWidth	`SetColumnWidth(aCol, aDIPs)` : ancho de una columna (índice de **Columns**) en píxeles independientes del dispositivo, usado en lugar de su **Width**. Un valor negativo restaura **Width**.
SaveToStream, SaveToFile	`SaveToStream(aData, aLines, aStream)` y `SaveToFile(aData, aLines, aFileName)` escriben el libro de `aData` . `aLines` proporciona la agrupación, el árbol y el estado expandido; null usa una vista de líneas propia, construida a partir del **GroupIndex** de las columnas.

Los errores lanzan **EsgcGridXLSXError**. `sgcGrid_XLSX.sgcGridExcelNumberFormat` devuelve el formato numérico de Excel usado para un tipo de datos y un **DisplayFormat**.

Ejemplo sin un control

Un programa de consola que escribe un libro agrupado con el primer grupo contraído, y un organigrama como árbol. Usa el método `CreateSales` del ejemplo de Agrupación y resúmenes de .NET:
Se escriben 100.000 filas en unos 1,4 segundos en .NET 8.

Consulte también

- Agrupación y resúmenes de .NET
- Filas de árbol y maestro detalle de .NET
- Clases de .NET
- Diferencias entre .NET y Delphi
- Demos de .NET
- Exportación a Excel

Demos de .NET

Las demos están en `.net\demos`, en la solución `sgcGrid_Demos.sln`. Cada demo tiene un README.md y un modificador `--test` que ejecuta una autocomprobación sin mostrar ventanas y termina con 0 si tiene éxito, como el modificador `/test` de las demos de Delphi.

Demo	Host	Muestra
LiveTicker	WinForms	1000 símbolos actualizados desde un subproceso de trabajo, con destello de cambios, colores de subida y bajada y **TailMode**.
LiveTickerWpf	WPF	El mismo ticker en **TsgcWpfGrid**.
LiveTickerAvalonia	Avalonia	El mismo ticker en **TsgcAvaloniaGrid**.
DataSetGrid	WinForms	Un DataTable enlazado mediante **TsgcGridSource_DataSet**, con importación y exportación de JSON y CSV.
Editing	WinForms	Edición en el lugar con **OnValidate**. Solo en la edición Registered.
GridAndChart	WinForms	Las filas y la selección de la cuadrícula controlan un gráfico de sgcCharts mediante **TsgcChartDataSource_Table**, dentro de mosaicos de sgcDashboard.
AskYourGrid	WinForms	**TsgcGridAIQuery** con el analizador basado en reglas. Solo en la edición Registered.
LiveFeed	WinForms	**TsgcGridFeedSource_WebSocket** alimentado por un **TsgcWebSocketServer** en el mismo proceso. Solo en la edición Registered.
Grouping	WinForms	100.000 pedidos agrupados por región y categoría, con el panel de grupos, resúmenes de grupo y de pie, dos columnas fijas y actualizaciones de precios en vivo desde un subproceso de trabajo.
GroupingWpf	WPF	La misma demo de agrupación con **TsgcWpfGrid**.
GroupingAvalonia	Avalonia	La misma demo de agrupación con **TsgcAvaloniaGrid**.
TreeAndDetail	WinForms	Un organigrama como filas de árbol, y clientes de un DataTable con una banda de detalle de sus pedidos desde un **TsgcGridDataStore**. Solo en la edición Registered.
ExcelExport	Consola	Escribe un libro de ventas agrupado y un organigrama como árbol con **TsgcGridXLSXWriter**, sin un control. Se ejecuta en Windows y Linux. Solo en la edición Registered.

Compilar las demos

El zip de las demos usa por defecto los paquetes NuGet de la edición Community. Compile las demos con los paquetes Registered estableciendo **SgcGridEdition**:

```
dotnet build sgcGrid_Demos.sln -p:SgcGridEdition=Registered
```

En la edición Community, las demos exclusivas de Registered siguen compilando y explican el límite cuando se ejecutan.

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Flujos en vivo en .NET
- Pregunte a su cuadrícula en .NET
- Agrupación y resúmenes de .NET
- Filas de árbol y maestro detalle de .NET
- Exportación a Excel de .NET
- Demos de Delphi

Solución de problemas

Undeclared identifier StartEdit, OnValidate u OnEdited

La edición Community se compila sin SGC_GRID_EDIT, por lo que los miembros de edición de las cuadrículas no existen y el código que los usa falla con E2003 Undeclared identifier. Un formulario guardado por una compilación Registered con eventos de edición lanza una excepción Error reading al cargarse. Elimine el código de edición, o use una compilación Registered o Trial. Consulte Ediciones.

File not found sgcGridWS_Source o sgcWebSocket

Los orígenes de flujo y la consulta de IA están en el puente sgcGridWS. La edición Community no lo distribuye. En las demás ediciones el puente necesita sgcWebSockets: añada sus fuentes a la ruta de búsqueda, o instale el paquete sgcWebSockets antes del paquete sgcGridWS de la misma versión. Consulte Plataformas.

El puente no se compila en Delphi 7

En Delphi 7 las unidades sgcGridWS se compilan contra sgcWebSockets, que necesita la versión de Indy para la que fue compilado. Instale la versión de Indy que espera su compilación de sgcWebSockets antes de compilar el puente.

La demo DataSetGrid de C++Builder falla en Win64

Win32 enlaza MIDAS de forma estática, pero C++Builder no tiene una biblioteca MIDAS estática para Win64. Copie midas.dll junto al ejecutable de 64 bits.

Los valores no cambian o las filas se duplican

UpdateRow localiza las filas por clave y las claves distinguen mayúsculas de minúsculas. Use la misma clave para AppendRow y UpdateRow. Con AutoInsert en False, la actualización de una clave inexistente se ignora. Los nombres de campo que no coinciden con ninguna columna también se ignoran.

OnFilterRow lee la fila equivocada

OnFilterRow recibe una fila del almacén. Léala con StoreCells o StoreKeyOf, no con Cells, que recibe filas de vista. Consulte Filtrado.

La cuadrícula se bloquea cuando un controlador de flujo toca un control

OnFilterRow y OnGetValue se ejecutan con el bloqueo de los datos retenido, y OnChange, OnRow y OnError de los flujos se ejecutan en el hilo escritor. No espere nunca al hilo principal allí y no toque nunca un control; use TThread.Queue en su lugar.

Los números muestran una coma o un punto de forma inesperada

Los valores se formatean con Data.FormatSettings, que siguen la configuración regional. Llame a Data.UseInvariantFormat para un separador decimal punto, o establezca FormatSettings.

Identificador no declarado Tree, Detail o SaveToXLSX

Las filas de árbol, las filas maestro detalle, TsgcGridDataStore y la exportación a Excel solo se compilan con SGC_GRID_ADVANCED, que la edición Community deja fuera. El código que los usa falla con E2003 Undeclared identifier, y un formulario guardado con propiedades Tree o Detail genera una excepción Error reading al cargarse. Vea Ediciones.

El árbol muestra una lista plana

Tree.KeyField y Tree.ParentField deben nombrar ambos columnas de la cuadrícula (pueden estar ocultas), y los datos no deben ser virtuales. Los valores de padre se comparan con los textos de clave distinguiendo mayúsculas; una fila cuyo padre no se encuentra se convierte en raíz. Vea Filas de árbol.

La agrupación no tiene efecto

La agrupación se ignora mientras el modo árbol está activo. Borre Tree.KeyField o Tree.ParentField para volver a agrupar.

No se muestran expansores de detalle

Una fila recibe un expansor solo cuando Detail.Active es True, Detail.MasterKeyField nombra una columna de la cuadrícula, hay un Detail.Source o un controlador OnGetDetailData asignado, y la clave maestra de la fila no está vacía. Un origen de datos virtual no muestra bandas. Vea Maestro detalle.

Una banda de detalle queda vacía

Detail.DetailKeyField debe nombrar una columna del origen, y sus valores deben coincidir con los textos de la clave maestra (distinguiendo mayúsculas; 5, 5.0 y '5' son la misma clave). Un origen virtual no da filas. Con OnGetDetailData y sin Source, asigne Detail.Columns, porque la banda toma sus columnas de ahí.

Faltan filas de detalle en el archivo de Excel

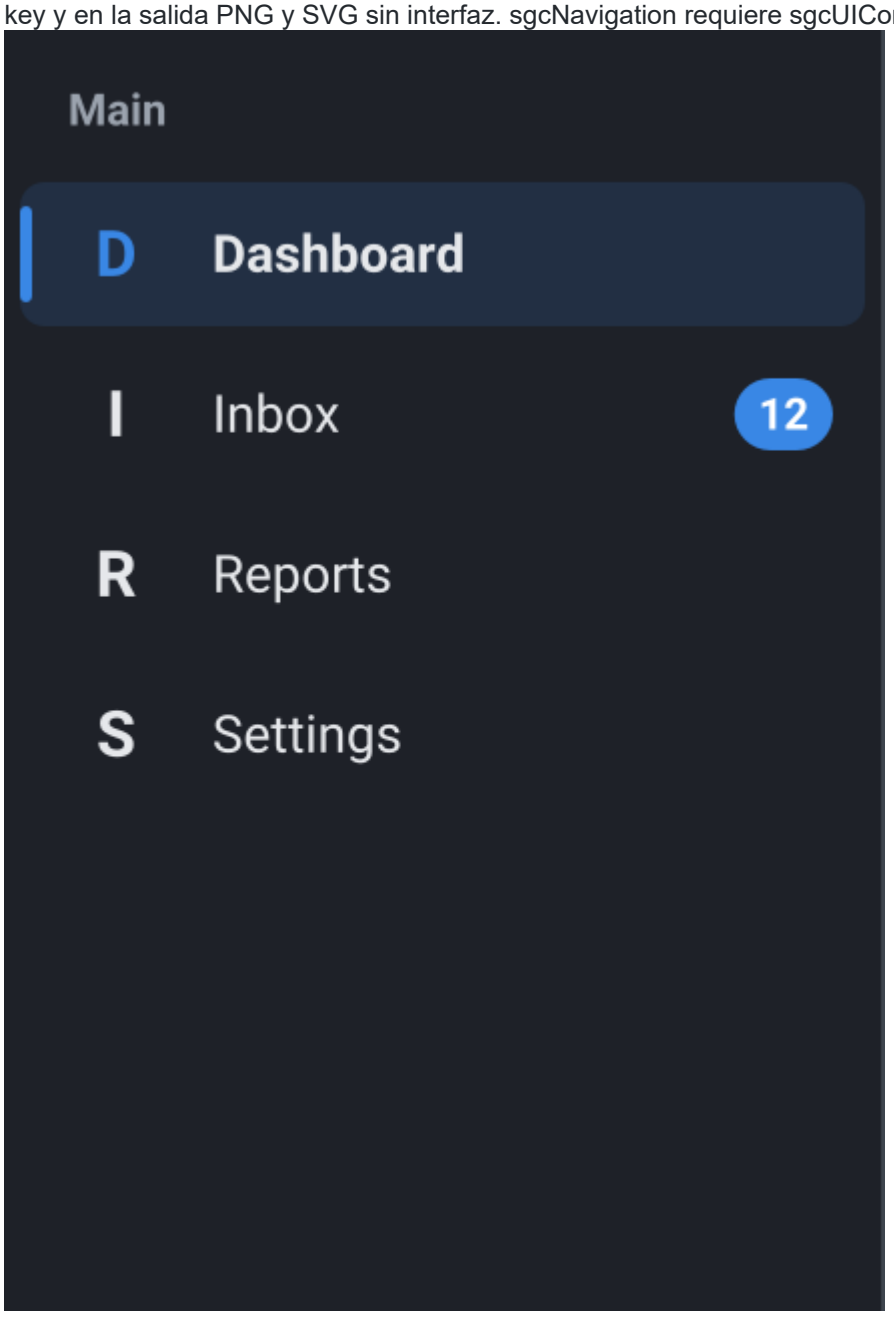
SaveToXLSX no escribe las bandas de detalle. Exporte el origen de detalle por separado, por ejemplo con un TsgcGridXLSXWriter sobre el Data del TsgcGridDataStore. Vea Exportación a Excel.

SaveToXLSX genera EsgcGridXLSXError

Una hoja de XLSX contiene como máximo 1.048.576 filas y 16.384 columnas, incluidas las filas de grupo y el encabezado. Filtre las filas u oculte columnas antes de la exportación. No se escribe nada cuando se genera la excepción.

Introducción

sgcNavigation es una barra de navegación nativa para Delphi y C++Builder, para VCL y FireMonkey. Un único componente, **TsgcVCLNavigationBar** o **TsgcFMXNavigationBar**, muestra las páginas de una aplicación como una lista vertical de elementos con iconos, títulos, insignias y cabeceras de grupo. En modo barra lateral es una lista plana que se contrae a un carril de iconos con una animación. En modo acordeón, las cabeceras de grupo expanden y contraen los elementos que tienen debajo.
sgcNavigation forma parte de la suite visual de eSeGeCe y enmarca los demás productos: una barra de navegación a la izquierda y, a la derecha, un dashboard **sgcDashboard**, una cuadrícula **sgcGrid** o una página con un **sgcToolbar**. Se dibuja con **sgcUICore**, el paquete de ejecución compartido con el lienzo, el rasterizador, las fuentes, el escritor SVG y los temas claro y oscuro, de modo que la barra tiene el mismo aspecto en VCL, en FireMonkey y en la salida PNG y SVG sin interfaz. sgcNavigation requiere sgcUICore, que se instala junto con él.



Ediciones

sgcNavigation se entrega como Registered, Trial y Community. La edición Community tiene todas las características excepto los vínculos de acción (Action) de los elementos. Consulte Ediciones.

Por dónde empezar

- Arquitectura: el modelo independiente del framework, el renderizador sobre el lienzo de sgcUICore y los controles VCL y FMX.
- Plataformas: las versiones de Delphi, C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX e Inicio rápido C++Builder: una primera barra de navegación en pocos minutos.
- Componentes: las barras de navegación, los elementos y el modelo.
- Modo barra lateral, Modo acordeón, Contraer y animación, Imágenes y glifos, Insignias, Teclado y foco, Vínculos de acción, Temas y Renderizado sin interfaz.
- Demos: los proyectos de demostración de Delphi y C++Builder.

Arquitectura

sgcNavigation se construye por capas. Cada capa solo depende de las inferiores, así que el modelo y el renderizador compilan sin VCL ni FireMonkey.

Capa	Unidad	Función
Modelo	sgcNavigation_Model	TsgcNavigationModel contiene los Items (una colección de TsgcNavigationItem), el modo, el estado contraído, los tamaños, el tema y el estado de interacción (elemento seleccionado, resaltado, pulsado y con foco). Su método Layout calcula la fila de cada elemento mostrado y HitTest encuentra el elemento situado bajo un punto.
Renderizador	sgcNavigation_Render	sgcNavigationRender dibuja un modelo en cualquier lienzo de sgcUICore. sgcNavigationSaveToPNG y sgcNavigationSaveToSVG lo usan para escribir una imagen sin ventana. Consulte Renderizado sin interfaz.
Control VCL	sgcNavigation_VCL	TsgcVCLNavigationBar posee un modelo, publica sus propiedades y lo dibuja mediante la base de controles VCL de sgcUICore. Añade el ratón, el teclado, los hints, la lista de imágenes y la animación de contracción.
Control FMX	sgcNavigation_FMX	TsgcFMXNavigationBar es el control FireMonkey con las mismas propiedades, métodos y eventos, además de entrada táctil.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme	El entorno de ejecución compartido: el lienzo vectorial y el rasterizador, la base de controles VCL y FMX con control de fotogramas (MaxFPS) y los tokens de los temas claro y oscuro.

Modelo y control

Las propiedades publicadas de una barra de navegación son las propiedades de su modelo, y la propiedad **Model** de la barra da acceso al resto: ToggleGroup, HitTest, NextVisible y el estado de interacción. Un cambio en el modelo programa un nuevo fotograma y nunca dibuja directamente, de modo que muchos cambios seguidos cuestan un único repintado.

Dibujo

El renderizador dibuja cada fila: el hueco del icono con una imagen o un glifo, el título ajustado al ancho libre, la insignia y los estados resaltado, pulsado, seleccionado, con foco y deshabilitado. Todos los tamaños son píxeles lógicos multiplicados por la escala del monitor, por lo que la barra se ve nítida con cualquier DPI. La barra VCL usa PPI / 96 como escala y la barra FMX la escala del formulario.

Consulte también

- Plataformas
- TsgcNavigationModel
- Temas

Ediciones

sgcNavigation se compila desde el mismo código fuente para todas las ediciones. La edición solo cambia las características que incluye una compilación.

Edición	Incluye
Registered	Todo: las barras de navegación VCL y FMX, los modos barra lateral y acordeón, el carril de iconos contraído con su animación, las imágenes, los glifos y las insignias, la navegación con teclado, los temas, los vínculos de acción (Action) de los elementos y la salida PNG y SVG sin interfaz.
Trial	Las mismas características que Registered. Una compilación con SGC_TRIAL muestra un aviso de versión de prueba cuando pasa la ventana de lanzamiento indicada en la compilación. Nada más cambia.
Community	Todo excepto los vínculos de acción (Action) de los elementos.

Qué omite Community

La edición Community compila con SGC_NAVIGATION_ACTIONS sin definir. En una compilación Community la propiedad **Action** de TsgcNavigationItem no existe, por lo que un elemento no se puede vincular a un TAction. Asigne en su lugar el **OnClick** del elemento o el evento **OnItemClick** de la barra.
La edición es una decisión en tiempo de compilación en sgcVer.inc (SGC_COMMUNITY, que también desactiva SGC_TRIAL), no un interruptor en tiempo de ejecución. El código o el archivo de formulario que usa Action no compila ni se carga con una compilación Community, consulte Solución de problemas.

Consulte también

- Vínculos de acción
- Plataformas

Instalación

sgcNavigation forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcNavigation** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcNavigation

sgcUICore, la base compartida de los productos de disposición y de sgcCharts, se incluye con sgcNavigation. No tiene casilla propia. El instalador la instala automáticamente siempre que se selecciona sgcNavigation. sgcNavigation no tiene paquete de puente de sgcWebSockets.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Navigation`	Las demos de sgcNavigation. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluido sgcNavigation.
- **Community**: el instalador gratuito Community de sgcUI incluye sgcNavigation. Sus productos son sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard, mientras que sgcPDF y sgcMaps no forman parte de Community. Consulte Ediciones para saber qué deja fuera la compilación Community de sgcNavigation.

Instalaciones independientes anteriores

Si el instalador encuentra una instalación independiente anterior de sgcUICore, como la que escribió el instalador independiente de sgcCharts, muestra las rutas de bibliotecas y los paquetes de diseño de esa instalación en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcNavigationD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcNavigation<V>` / `dclsgcNavigation<V>`	Los componentes de sgcNavigation (tiempo de ejecución / tiempo de diseño).

Compílelos en orden de dependencia: `sgcUICore`, después `sgcNavigation`, después `dclsgcNavigation`. Después instale el paquete de diseño `dclsgcNavigation` y añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` correspondiente a su versión, por ejemplo `sgcNavigationC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

.NET

No cambia nada para .NET. sgcNavigation se sigue distribuyendo como sus propios paquetes NuGet (`esegece.sgcNavigation` y sus paquetes complementarios). Consulte la instalación de .NET.

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13. Los paquetes generan las cabeceras .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	TsgcVCLNavigationBar en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	TsgcFMXNavigationBar desde Delphi XE6 (SGC_NAVIGATION_FMX), en Win32, Win64, macOS, iOS y Android, y en Linux con FMX para Linux (FMXLinux). La propiedad Images requiere XE8 o posterior (FMX.ImgList), y los hints de los elementos requieren 10.1 Berlin o posterior.
Linux64	Las unidades independientes del framework, el modelo y el renderizador con la salida PNG y SVG sin interfaz, desde Delphi 10.2 (SGC_NAVIGATION_LINUX).
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, y la salida headless en PNG y SVG, en .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Paquetes

sgcNavigation incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcNavigationD13.dpk para Delphi 13 y sgcNavigationC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcNavigation	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, sgcUICore
sgcNavigationVCL	Runtime, VCL	sgcNavigation_VCL (TsgcVCLNavigationBar)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcNavigation
sgcNavigationFMX	Runtime, FireMonkey, desde XE6	sgcNavigation_FMX (TsgcFMXNavigationBar)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcNavigation
dclsgcNavigation	Tiempo de diseño, VCL	Registra TsgcVCLNavigationBar en la página de paleta **SGC Navigation**.	designide, rtl, vcl, sgcNavigation, sgcNavigationVCL
dclsgcNavigationFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXNavigationBar en la página de paleta **SGC Navigation**.	designide, rtl, fmx, sgcNavigation, sgcNavigationFMX, dclsgcNavigation

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcNavigationFMX requiere dclsgcNavigation, así que instale primero dclsgcNavigation.

Versiones del IDE y plataformas de destino

Delphi	sgcNavigation	sgcNavigationVCL	sgcNavigationFMX	dclsgcNavigation, dclsgcNavigationFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcNavigation, sgcNavigationVCL	sgcNavigationFMX	dclsgcNavigation, dclsgcNavigationFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcNavigation)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcNavigation)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcNavigationC13.cbproj, sgcNavigationVCLC13.cbproj, sgcNavigationFMXC13.cbproj, dclsgcNavigationC13.cbproj y dclsgcNavigationFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXNavigationBar) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcNavigationVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcNavigation_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcNavigation se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa las unidades independientes del framework. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.
La barra VCL solo aparece en la paleta de un formulario VCL y la barra FMX solo en la de un formulario FireMonkey. La propiedad **Items** se edita con el editor de colecciones estándar.

Consulte también

- Ediciones
- Arquitectura
- Solución de problemas

Inicio rápido VCL

En tiempo de diseño

- Coloque un **TsgcVCLNavigationBar** de la página de paleta **SGC Navigation** en un formulario. Está alineado a la izquierda por defecto.
- Haga doble clic en **Items** en el Object Inspector y añada los elementos. Establezca **Caption** y **Glyph** (una letra o un símbolo corto) en cada elemento, y establezca **IsGroup** a True en un elemento que sea una cabecera de grupo.
- Coloque un TPanel o un TPageControl en el resto del formulario con **Align** establecido a alClient.
- Escriba un manejador de **OnItemClick** que muestre la página del elemento pulsado, o asigne el **OnClick** de cada elemento.
- Establezca **Theme** a utDark para la paleta oscura y **Collapsed** a True para ver el carril de iconos.

En código

La misma barra creada en tiempo de ejecución. El programa compila tal cual con `sgcUI\Source` en la ruta de búsqueda.

```
uses
  Forms, Classes, Controls, StdCtrls, ExtCtrls,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oPanel: TPanel;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcNavigation VCL Quick Start';
  Width := 900;
  Height := 560;

  // the navigation bar, aligned to the left of the form
  FNav := TsgcVCLNavigationBar.Create(Self);
  FNav.Parent := Self;
  FNav.Align := alLeft;
  FNav.Mode := nbmSidebar;

  // a group header, then the items: Caption, Glyph (drawn in the icon slot)
  FNav.Items.AddItem('Main', '', True);
  FNav.Items.AddItem('Dashboard', 'D');
  FNav.Items.AddItem('Inbox', 'I').Badge := '12';
  FNav.Items.AddItem('Reports', 'R');
  FNav.Items.AddItem('System', '', True);
  FNav.Items.AddItem('Settings', 'S').Hint := 'Application settings';
  FNav.SelectedIndex := 1;
  FNav.OnItemClick := OnNavItemClick;

  // the content area on the right
  oPanel := TPanel.Create(Self);
  oPanel.Parent := Self;
  oPanel.Align := alClient;
  FTitle := TLabel.Create(Self);
  FTitle.Parent := oPanel;
  FTitle.Left := 16;
  FTitle.Top := 16;
  FTitle.Caption := 'Dashboard';

  FNav.Theme := utDark;
end;
```

```
procedure TfrmMain.OnNavItemClick(Sender: TObject; aItem: TsgcNavigationItem);
begin
  FTitle.Caption := aItem.Caption;
end;
```

Consulte también

- TsgcVCLNavigationBar
- TsgcNavigationItem
- Modo barra lateral
- Contraer y animación
- Temas

Inicio rápido FMX

El control FireMonkey requiere Delphi XE6 o posterior. Se usa exactamente igual que el de VCL: coloque un **TsgcFMXNavigationBar**, que está alineado a la izquierda por defecto, rellene **Items** con el editor de colecciones y gestione **OnItemClick**. Un toque sobre una fila funciona como un clic.

En código

```
uses
  System.Classes, FMX.Forms, FMX.Types, FMX.StdCtrls,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_FMX;

constructor TfrmMain.Create(aOwner: TComponent);
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcNavigation FMX Quick Start';
  Width := 900;
  Height := 560;

  FNav := TsgcFMXNavigationBar.Create(Self);
  FNav.Parent := Self;
  FNav.Align := TAlignLayout.Left;
  FNav.Items.AddItem('Main', '', True);
  FNav.Items.AddItem('Dashboard', 'D');
  FNav.Items.AddItem('Inbox', 'I').Badge := '12';
  FNav.Items.AddItem('Settings', 'S');
  FNav.SelectedIndex := 1;
  FNav.OnItemClick := OnNavItemClick;

  FTitle := TLabel.Create(Self);
  FTitle.Parent := Self;
  FTitle.Align := TAlignLayout.Client;
  FTitle.Text := 'Dashboard';

  FNav.Theme := utDark;
end;

procedure TfrmMain.OnNavItemClick(Sender: TObject; aItem: TsgcNavigationItem);
begin
  FTitle.Text := aItem.Caption;
end;
```

Consulte también

- TsgcFMXNavigationBar
- TsgcNavigationItem
- Plataformas

Inicio rápido C++Builder

Instale los paquetes de sgcNavigation para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de inclusión y de bibliotecas del proyecto, y use los componentes de la página de paleta **SGC Navigation** exactamente igual que en Delphi. El instalador añade ambas rutas al IDE por usted. Los cabeceras .hpp y los archivos .lib generados de sgcNavigation y sgcUICore están en esa única carpeta, por lo que sgcUICore no necesita una ruta propia.

En código

OnItemClick es un evento de tipo closure, por lo que su manejador es una función miembro __fastcall de una clase.

Consulte también

- Inicio rápido VCL
- Demos
- Plataformas

Demos

Las demos están en la carpeta **Demos** de la instalación, con los proyectos de Delphi en Demos\Delphi y las versiones para C++Builder de las demos VCL en Demos\CBuilder. Cada carpeta de demo tiene un readme.html que explica qué hace la demo y cómo funciona.

Demo	Componentes	Qué muestra
SidebarApp	TsgcVCLNavigationBar, TsgcVCLDashboard, TsgcVCLKPICard	Una barra de navegación en modo barra lateral con iconos, insignias, un elemento deshabilitado y una contracción animada, que controla las páginas de un TPageControl, una de ellas un dashboard sgcDashboard. Delphi y C++Builder, Win32 y Win64. Requiere sgcDashboard.
Accordion	TsgcVCLNavigationBar, TActionList	Una barra de navegación en modo acordeón con tres grupos, un interruptor de expansión simple o múltiple y elementos vinculados a las acciones de un TActionList. Delphi y C++Builder, Win32 y Win64. Los vínculos de acción no forman parte de la edición Community, por lo que esta demo no compila con ella.
SidebarFMX	TsgcFMXNavigationBar, TsgcFMXDashboard, TsgcFMXKPICard	La versión FireMonkey de SidebarApp: glifos, insignias, un elemento deshabilitado y una contracción animada, que controla las pestañas de un TTabControl, una de ellas un dashboard sgcDashboard. Delphi, Win32 y Win64. Requiere sgcDashboard.

Consulte también

- Inicio rápido VCL
- Modo acordeón
- Vínculos de acción

TsgcVCLNavigationBar

Unidad: sgcNavigation_VCL
TsgcVCLNavigationBar es la barra de navegación VCL, solo para Windows. Está alineada a la izquierda por defecto y muestra sus Items como filas, en modo barra lateral o en modo acordeón. Gestiona el ratón, la rueda, el teclado y los hints, y se desplaza cuando las filas son más altas que el control. Todos los tamaños son píxeles lógicos multiplicados por PPI / 96.

Propiedades

Propiedad	Tipo	Descripción
Items	TsgcNavigationItems	Los elementos y las cabeceras de grupo, en orden de presentación. Consulte TsgcNavigationItems.
Mode	TsgcNavigationMode	nbmSidebar (por defecto), una lista plana, o nbmAccordion, grupos que se expanden y se contraen.
Collapsed	Boolean	Muestra la barra como un carril de iconos de ancho CollapsedWidth en lugar de ExpandedWidth. Por defecto False.
ExpandedWidth	Integer	Ancho de la barra expandida en píxeles lógicos. Por defecto 220.
CollapsedWidth	Integer	Ancho del carril de iconos en píxeles lógicos. Por defecto 56.
ItemHeight	Integer	Alto de una fila de elemento en píxeles lógicos. Por defecto 40.
GroupHeight	Integer	Alto de una fila de cabecera de grupo en píxeles lógicos. Por defecto 32.
Indent	Integer	Sangría de los elementos de un grupo en modo acordeón, en píxeles lógicos. Por defecto 16.
AllowMultiExpand	Boolean	En modo acordeón, permite que más de un grupo esté expandido. Con False (por defecto), expandir un grupo contrae los demás.
SelectedIndex	Integer	Índice del elemento seleccionado, -1 (por defecto) si no hay ninguno. Un índice fuera de rango se convierte en -1.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom, que usa los colores claros.
AccentColor	TsgcUIColor	Color de la selección, del foco y de las insignias como $AARRGGBB. 0 (por defecto) usa el acento del tema.
Images	TCustomImageList	La lista de imágenes del ImageIndex del elemento. Las imágenes se dibujan con su propio tamaño.
AnimateCollapse	Boolean	Anima el ancho durante unos 150 ms cuando Collapsed cambia en tiempo de ejecución. Por defecto True.
Model	TsgcNavigationModel (public, solo lectura)	El modelo de la barra. Consulte TsgcNavigationModel.

ScrollY	Integer (public, solo lectura)	El desplazamiento vertical en píxeles de dispositivo.

También se publican los miembros habituales de un control: **Align** (por defecto alLeft), **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop** (por defecto True), **Visible**, **OnClick**, **OnContextPopup**, **OnEnter**, **OnExit**, **OnKeyDown**, **OnKeyUp**, **OnMouseDown**, **OnMouseMove**, **OnMouseUp** y **OnResize**.

Métodos

Método	Devuelve	Descripción
ItemAtPos(X, Y: Integer)	Integer	El elemento situado en un punto del control, -1 si no hay ninguno. Una sobrecarga devuelve también la parte situada bajo el punto, npItem o npExpander.
ItemRect(aIndex: Integer)	TRect	La fila de un elemento en coordenadas del control, con el desplazamiento aplicado, o un rectángulo vacío cuando el elemento no se muestra.
ScrollIntoView(aIndex: Integer)		Desplaza la barra para que la fila del elemento sea visible.
InitiateAction		Permite que cada elemento con una Action se actualice. El framework lo llama en reposo.

Eventos

Evento	Tipo	Descripción
OnItemClick	TsgcNavigationItemEvent	Se hizo clic, se tocó o se activó con Enter o Espacio un elemento habilitado. El elemento ya está seleccionado. El OnClick del elemento o su Action se ejecuta después de este evento.
OnSelectionChange	TNotifyEvent	SelectedIndex cambió, por el usuario o desde código.
OnCollapsedChange	TNotifyEvent	Collapsed cambió.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ExtCtrls,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oPanel: TPanel;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcNavigation VCL Quick Start';
  Width := 900;
  Height := 560;
```

```
  // the navigation bar, aligned to the left of the form
  FNav := TsgcVCLNavigationBar.Create(Self);
  FNav.Parent := Self;
  FNav.Align := alLeft;
  FNav.Mode := nbmSidebar;

  // a group header, then the items: Caption, Glyph (drawn in the icon slot)
  FNav.Items.AddItem('Main', '', True);
  FNav.Items.AddItem('Dashboard', 'D');
  FNav.Items.AddItem('Inbox', 'I').Badge := '12';
  FNav.Items.AddItem('Reports', 'R');
  FNav.Items.AddItem('System', '', True);
  FNav.Items.AddItem('Settings', 'S').Hint := 'Application settings';
  FNav.SelectedIndex := 1;
  FNav.OnItemClick := OnNavItemClick;

  // the content area on the right
  oPanel := TPanel.Create(Self);
  oPanel.Parent := Self;
  oPanel.Align := alClient;
  FTitle := TLabel.Create(Self);
  FTitle.Parent := oPanel;
  FTitle.Left := 16;
  FTitle.Top := 16;
  FTitle.Caption := 'Dashboard';

  FNav.Theme := utDark;
end;

procedure TfrmMain.OnNavItemClick(Sender: TObject; aItem: TsgcNavigationItem);
begin
  FTitle.Caption := aItem.Caption;
end;
```

Consulte también

- TsgcFMXNavigationBar
- TsgcNavigationItem
- TsgcNavigationModel
- Teclado y foco

TsgcFMXNavigationBar

Unidad: sgcNavigation_FMX
TsgcFMXNavigationBar es la barra de navegación FireMonkey, desde Delphi XE6, para Windows, macOS, iOS, Android y Linux con FMXLinux. Tiene las propiedades, los métodos y los eventos de TsgcVCLNavigationBar y añade entrada táctil: un toque sobre una fila funciona como un clic. Los tamaños son unidades lógicas multiplicadas por la escala del formulario. Desde 10.1 Berlin, el Hint del control sigue al elemento situado bajo el cursor.

Propiedades

Propiedad	Tipo	Descripción
Items	TsgcNavigationItems	Los elementos y las cabeceras de grupo, en orden de presentación. Consulte TsgcNavigationItems.
Mode	TsgcNavigationMode	nbmSidebar (por defecto), una lista plana, o nbmAccordion, grupos que se expanden y se contraen.
Collapsed	Boolean	Muestra la barra como un carril de iconos de ancho CollapsedWidth en lugar de ExpandedWidth. Por defecto False.
ExpandedWidth	Integer	Ancho de la barra expandida en píxeles lógicos. Por defecto 220.
CollapsedWidth	Integer	Ancho del carril de iconos en píxeles lógicos. Por defecto 56.
ItemHeight	Integer	Alto de una fila de elemento en píxeles lógicos. Por defecto 40.
GroupHeight	Integer	Alto de una fila de cabecera de grupo en píxeles lógicos. Por defecto 32.
Indent	Integer	Sangría de los elementos de un grupo en modo acordeón, en píxeles lógicos. Por defecto 16.
AllowMultiExpand	Boolean	En modo acordeón, permite que más de un grupo esté expandido. Con False (por defecto), expandir un grupo contrae los demás.
SelectedIndex	Integer	Índice del elemento seleccionado, -1 (por defecto) si no hay ninguno. Un índice fuera de rango se convierte en -1.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom, que usa los colores claros.
AccentColor	TsgcUIColor	Color de la selección, del foco y de las insignias como $AARRGGBB. 0 (por defecto) usa el acento del tema.
Images	TCustomImageList	La lista de imágenes del ImageIndex del elemento. En FMX requiere XE8 o posterior y las imágenes se dibujan a 16 x 16 píxeles lógicos.
AnimateCollapse	Boolean	Anima el ancho durante unos 150 ms cuando Collapsed cambia en tiempo de ejecución. Por defecto True.
Model	TsgcNavigationModel (public, solo lectura)	El modelo de la barra. Consulte TsgcNavigationModel.

ScrollY	Integer (public, solo lectura)	El desplazamiento vertical en píxeles de dispositivo.

También se publican los miembros habituales de un control FMX, entre ellos **Align** (por defecto Left), **Anchors**, **CanFocus** (por defecto True), **Enabled**, **HitTest**, **Margins**, **Padding**, **Opacity**, **PopupMenu**, **Position**, **Width**, **Height**, **TabOrder**, **Visible**, **ShowHint** y **TabStop** (10.1 y posteriores), y los eventos OnClick, OnEnter, OnExit, OnKey, OnMouse, OnPaint y OnResize.

Métodos

Método	Devuelve	Descripción
ItemAtPos(X, Y: Single)	Integer	El elemento situado en un punto del control, -1 si no hay ninguno. Una sobrecarga devuelve también la parte situada bajo el punto, npItem o npExpander.
ItemRect(aIndex: Integer)	TRectF	La fila de un elemento en coordenadas del control, con el desplazamiento aplicado, o un rectángulo vacío cuando el elemento no se muestra.
ScrollIntoView(aIndex: Integer)		Desplaza la barra para que la fila del elemento sea visible.
InitiateAction		Permite que cada elemento con una Action se actualice. El framework lo llama en reposo.

Eventos

Evento	Tipo	Descripción
OnItemClick	TsgcNavigationItemEvent	Se hizo clic, se tocó o se activó con Enter o Espacio un elemento habilitado. El elemento ya está seleccionado. El OnClick del elemento o su Action se ejecuta después de este evento.
OnSelectionChange	TNotifyEvent	SelectedIndex cambió, por el usuario o desde código.
OnCollapsedChange	TNotifyEvent	Collapsed cambió.

Ejemplo

```
uses
  System.Classes, FMX.Forms, FMX.Types, FMX.StdCtrls,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_FMX;

constructor TfrmMain.Create(aOwner: TComponent);
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcNavigation FMX Quick Start';
  Width := 900;
  Height := 560;

  FNav := TsgcFMXNavigationBar.Create(Self);
```

```
  FNav.Parent := Self;
  FNav.Align := TAlignLayout.Left;
  FNav.Items.AddItem('Main', '', True);
  FNav.Items.AddItem('Dashboard', 'D');
  FNav.Items.AddItem('Inbox', 'I').Badge := '12';
  FNav.Items.AddItem('Settings', 'S');
  FNav.SelectedIndex := 1;
  FNav.OnItemClick := OnNavItemClick;

  FTitle := TLabel.Create(Self);
  FTitle.Parent := Self;
  FTitle.Align := TAlignLayout.Client;
  FTitle.Text := 'Dashboard';

  FNav.Theme := utDark;
end;

procedure TfrmMain.OnNavItemClick(Sender: TObject; aItem: TsgcNavigationItem);
begin
  FTitle.Text := aItem.Caption;
end;
```

Consulte también

- TsgcVCLNavigationBar
- TsgcNavigationItem
- Plataformas

TsgcNavigationItem

Unidad: sgcNavigation_Model
TsgcNavigationItem es una fila de una barra de navegación: una página de la aplicación, o una cabecera de grupo cuando IsGroup es True. Los elementos viven en la colección Items de la barra.

Propiedades

Propiedad	Tipo	Descripción
Caption	string	El texto de la fila. Se oculta en el carril de iconos contraído.
Glyph	string	Un texto o símbolo corto, por ejemplo una letra, dibujado centrado en el hueco del icono cuando el elemento no tiene imagen.
ImageIndex	Integer	Índice de la imagen en Images de la barra. Por defecto -1, sin imagen.
Badge	string	Un texto corto dibujado como una píldora de acento a la derecha de la fila, un punto sobre el icono en el carril. Vacío (por defecto) no dibuja insignia. Consulte Insignias.
IsGroup	Boolean	Convierte el elemento en una cabecera de grupo. Una cabecera nunca se selecciona. En modo acordeón expande y contrae los elementos que tiene debajo hasta la próxima cabecera. Por defecto False.
Expanded	Boolean	Indica si los elementos de una cabecera de grupo se muestran en modo acordeón. Por defecto True.
Enabled	Boolean	Un elemento deshabilitado se dibuja atenuado y no se puede seleccionar ni pulsar. Por defecto True.
Visible	Boolean	Un elemento oculto no recibe fila. Por defecto True.
Hint	string	El hint de la fila. En el carril contraído, un elemento sin Hint muestra su Caption como hint.
Tag	NativeInt	Un valor libre para la aplicación, Integer antes de XE2.
Action	TBasicAction	Una acción vinculada. Caption, Enabled, Visible, Hint e ImageIndex la siguen y un clic la ejecuta, salvo que OnClick esté asignado. No está en la edición Community. Consulte Vínculos de acción.
BoundsRect	TsgcUIRect (public, solo lectura)	La fila de la última disposición en píxeles de dispositivo, en coordenadas de contenido antes del desplazamiento.
Shown	Boolean (public, solo lectura)	Indica si la última disposición asignó una fila al elemento.
Separator	Boolean (public,	Indica si la fila es una línea separadora de un grupo en el carril.

	solo lectura)	
Level	Integer (public, solo lectura)	El nivel de sangría de la fila, 1 para los elementos de un grupo de acordeón.

Métodos

Método	Devuelve	Descripción
Click(aComponent: TComponent = nil)		Ejecuta el elemento: su OnClick si está asignado y, si no, su Action, ejecutada con aComponent como ActionComponent. La barra lo llama después de OnItemClick.
InitiateAction		Permite que la Action vinculada se actualice. No hace nada sin una Action.

Eventos

Evento	Tipo	Descripción
OnClick	TNotifyEvent	El elemento se activó. Sender es el elemento.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupItems;
var
  oItem: TsgcNavigationItem;
begin
  FNav.Items.Clear;
  FNav.Items.AddItem('Workspace', '', True);
  FNav.Items.AddItem('Dashboard', 'D');

  oItem := FNav.Items.Add;
  oItem.Caption := 'Inbox';
  oItem.Glyph := 'I';
  oItem.Badge := '12';
  oItem.Hint := '12 unread messages';

  oItem := FNav.Items.AddItem('Billing', 'B');
  oItem.Enabled := False;
  oItem.Hint := 'Requires an administrator account';

  oItem := FNav.Items.AddItem('Settings', 'S');
  oItem.Tag := 100;
  oItem.OnClick := OnSettingsClick;
end;

procedure TfrmMain.OnSettingsClick(Sender: TObject);
begin
  // Sender is the item
  FStatus.Caption := 'Settings, tag ' +
```

```
    IntToStr((Sender as TsgcNavigationItem).Tag);
end;
```

Consulte también

- TsgcNavigationItems
- Imágenes y glifos
- Insignias
- Vínculos de acción

TsgcNavigationItems

Unidad: sgcNavigation_Model
TsgcNavigationItems es la colección de TsgcNavigationItem de una barra de navegación, un TOwnedCollection, por lo que los elementos se editan en tiempo de diseño con el editor de colecciones estándar y se guardan en el archivo de formulario. El orden de la colección es el orden de las filas. Una cabecera de grupo posee los elementos que la siguen hasta la próxima cabecera.

Métodos

Método	Devuelve	Descripción
Add	TsgcNavigationItem	Añade un elemento vacío al final.
AddItem(const aCaption: string; const aGlyph: string = ''; aIsGroup: Boolean = False)	TsgcNavigationItem	Añade un elemento, o una cabecera de grupo cuando aIsGroup es True, con su Caption y su Glyph, y lo devuelve.
Items[Index: Integer]	TsgcNavigationItem	El elemento en un índice. Es la propiedad predeterminada, así que Nav.Items[2] es el tercer elemento.

Los miembros heredados de TCollection también están disponibles: **Count**, **Clear**, **Delete**, **Insert**, **BeginUpdate** y **EndUpdate**. Envuelva muchos cambios en BeginUpdate y EndUpdate para calcular la disposición de la barra una sola vez.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupAccordion;
begin
  FNav.Items.BeginUpdate;
  try
    FNav.Items.Clear;
    // a group header owns the items that follow it until the next header
    FNav.Items.AddItem('Sales', '', True);
    FNav.Items.AddItem('Orders', 'O');
    FNav.Items.AddItem('Customers', 'C').Badge := '3';
    FNav.Items.AddItem('Inventory', '', True).Expanded := False;
    FNav.Items.AddItem('Products', 'P');
    FNav.Items.AddItem('Stock', 'S');
    FNav.Items.AddItem('Admin', '', True).Expanded := False;
    FNav.Items.AddItem('Users', 'U');
  finally
    FNav.Items.EndUpdate;
  end;
  FNav.Mode := nbmAccordion;
  // False: expanding a group closes the others
  FNav.AllowMultiExpand := False;
  // expand the Inventory group (index 3) from code
  FNav.Model.ToggleGroup(3);
end;
```

Consulte también

- TsgcNavigationItem
- Modo acordeón

TsgcNavigationModel

Unidad: sgcNavigation_Model
TsgcNavigationModel es el modelo independiente del framework de una barra de navegación. Cada barra posee uno, publicado mediante su propiedad **Model**, y también se puede crear un modelo por sí solo, sin control, para renderizar una barra como imagen o para probar la disposición. No tiene dependencias de VCL ni de FMX y compila en todas las plataformas. Está pensado para usarse desde el hilo principal.

Propiedades

Las propiedades publicadas son las de la barra: Items, Mode, Collapsed, ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight, Indent, AllowMultiExpand, SelectedIndex, Theme y AccentColor, consulte TsgcVCLNavigationBar. El modelo añade estas propiedades public:

Propiedad	Tipo	Descripción
HotIndex	Integer	El elemento situado bajo el ratón, -1 si no hay ninguno.
PressedIndex	Integer	El elemento que se está pulsando, -1 si no hay ninguno.
FocusedIndex	Integer	El elemento con el foco del teclado, -1 si no hay ninguno.
ShowFocus	Boolean	Dibuja el rectángulo de foco. El control lo establece mientras se usa el teclado.
LayoutWidth	Integer (solo lectura)	El ancho del último Layout, en píxeles de dispositivo.
LayoutScale	Double (solo lectura)	La escala del último Layout.

Métodos

Método	Devuelve	Descripción
Layout(aWidth: Integer; aScale: Double = 1)		Calcula las filas de una barra de aWidth píxeles de dispositivo de ancho. aScale multiplica cada tamaño lógico.
ContentHeight	Double	Alto de las filas más el relleno del último Layout, en píxeles de dispositivo.
HitTest(aX, aY: Double; out aPart: TsgcNavigationPart)	Integer	El elemento situado en un punto, en coordenadas de contenido del último Layout, -1 si no hay ninguno. aPart es npItem, o npExpander sobre el expansor de un grupo de acordeón. Una sobrecarga no tiene aPart. Se informa de los elementos deshabilitados, no de un separador del carril.
IsItemShown(aIndex: Integer)	Boolean	Indica si el elemento recibe una fila con el Mode, el Collapsed y el estado de grupo actuales.
IsItemFocusable(aIndex: Integer)	Boolean	Indica si el teclado puede detenerse en el elemento: mostrado, habilitado y que no sea una cabecera de grupo, salvo en el acordeón expandido.

NextVisible(aIndex: Integer)	Integer	El siguiente elemento que puede recibir el foco, -1 si no hay ninguno. NextVisible(-1) es el primero.
PrevVisible(aIndex: Integer)	Integer	El elemento anterior que puede recibir el foco, -1 si no hay ninguno. PrevVisible(Items.Count) es el último.
GroupOf(aIndex: Integer)	Integer	La cabecera de grupo a la que pertenece un elemento, -1 cuando el elemento es un grupo o ningún grupo lo precede.
ToggleGroup(aIndex: Integer)		Expande o contrae una cabecera de grupo. Expandir contrae los demás grupos cuando AllowMultiExpand es False.
CurrentWidth	Integer	El ancho lógico para el estado actual de Collapsed, ExpandedWidth o CollapsedWidth.
GetAccent	TsgcUIColor	El color de acento en uso: AccentColor, o el acento del tema cuando es 0.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El modelo cambió y necesita repintarse. La barra lo asigna, así que úselo solo en un modelo creado por sí solo.

Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_Render;

var
  oModel: TsgcNavigationModel;
  vIndex: Integer;
  vPart: TsgcNavigationPart;
begin
  oModel := TsgcNavigationModel.Create;
  try
    oModel.Items.AddItem('Main', '', True);
    oModel.Items.AddItem('Dashboard', 'D');
    oModel.Items.AddItem('Inbox', 'I').Badge := '12';
    oModel.Items.AddItem('Reports', 'R');
    oModel.Items.AddItem('Settings', 'S');
    oModel.SelectedIndex := 1;
    oModel.Theme := utDark;

    // 220 x 320 logical pixels, written at scale 2 as 440 x 640 pixels
    sgcNavigationSaveToPNG(oModel, 'navigation.png', 220, 320, 2);
    sgcNavigationSaveToSVG(oModel, 'navigation.svg', 220, 320);

    // the icon rail of the same bar
    oModel.Collapsed := True;
    sgcNavigationSaveToPNG(oModel, 'navigation_rail.png',
      oModel.CurrentWidth, 320);

    // the row under a point, after a Layout at the same width and scale
    oModel.Collapsed := False;
    oModel.Layout(220, 1);
    vIndex := oModel.HitTest(40, 60, vPart);
    if vIndex >= 0 then
```

```
      WriteLn('Hit: ', oModel.Items[vIndex].Caption);
    WriteLn('Content height: ', Round(oModel.ContentHeight));
  finally
    oModel.Free;
  end;
end;
```

Consulte también

- Renderizado sin interfaz
- Arquitectura
- TsgcVCLNavigationBar

Modo barra lateral

El modo barra lateral, **Mode** = nbmSidebar, es el predeterminado. La barra muestra cada elemento visible como una fila, apiladas desde arriba. Una cabecera de grupo es una etiqueta estática de color atenuado que titula los elementos que tiene debajo, nunca se selecciona y un clic sobre ella no hace nada.

- Una fila de elemento mide ItemHeight píxeles de alto y una fila de cabecera de grupo GroupHeight píxeles.
- Un clic o un toque sobre un elemento habilitado lo selecciona, dispara OnItemClick y después ejecuta el OnClick del elemento o su Action.
- La fila seleccionada recibe un fondo teñido con el color de acento y una barra de acento de 3 píxeles en su borde izquierdo.
- Cuando las filas son más altas que el control, la rueda del ratón desplaza la barra, y ScrollIntoView muestra un elemento dado.
- Collapsed convierte la barra lateral en un carril de iconos, consulte Contraer y animación.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ExtCtrls,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

constructor TfrmMain.Create(aOwner: TComponent);
var
  oPanel: TPanel;
begin
  inherited CreateNew(aOwner);
  Caption := 'sgcNavigation VCL Quick Start';
  Width := 900;
  Height := 560;

  // the navigation bar, aligned to the left of the form
  FNav := TsgcVCLNavigationBar.Create(Self);
  FNav.Parent := Self;
  FNav.Align := alLeft;
  FNav.Mode := nbmSidebar;

  // a group header, then the items: Caption, Glyph (drawn in the icon slot)
  FNav.Items.AddItem('Main', '', True);
  FNav.Items.AddItem('Dashboard', 'D');
  FNav.Items.AddItem('Inbox', 'I').Badge := '12';
  FNav.Items.AddItem('Reports', 'R');
  FNav.Items.AddItem('System', '', True);
  FNav.Items.AddItem('Settings', 'S').Hint := 'Application settings';
  FNav.SelectedIndex := 1;
  FNav.OnItemClick := OnNavItemClick;

  // the content area on the right
  oPanel := TPanel.Create(Self);
  oPanel.Parent := Self;
  oPanel.Align := alClient;
  FTitle := TLabel.Create(Self);
  FTitle.Parent := oPanel;
  FTitle.Left := 16;
  FTitle.Top := 16;
  FTitle.Caption := 'Dashboard';

  FNav.Theme := utDark;
end;

procedure TfrmMain.OnNavItemClick(Sender: TObject; aItem: TsgcNavigationItem);
begin
```

```
  FTitle.Caption := aItem.Caption;
end;
```

Consulte también

- Modo acordeón
- Contraer y animación
- TsgcVCLNavigationBar

Modo acordeón

En modo acordeón, **Mode** = nbmAccordion, una cabecera de grupo expande y contrae los elementos que la siguen hasta la próxima cabecera. Los elementos de un grupo solo se muestran mientras el grupo es visible y su propiedad **Expanded** es True, y se sangran con **Indent**. La cabecera tiene un expansor en forma de chevrón en su borde derecho.

- Un clic en la fila de la cabecera o en su expansor alterna el grupo. El teclado hace lo mismo con Enter o Espacio, Izquierda contrae y Derecha expande el grupo con foco.
- Con **AllowMultiExpand** False (por defecto), expandir un grupo contrae los demás, de modo que solo hay un grupo abierto a la vez. Establézcalo a True para dejar abiertos varios grupos.
- En código, **Model.ToggleGroup** alterna un grupo con la misma regla, y establecer **Expanded** de una cabecera la abre o la cierra sin tocar las demás.
- Los elementos anteriores a la primera cabecera de grupo no pertenecen a ningún grupo y siempre se muestran.
- En el carril contraído los grupos no se alternan: se muestra cada elemento visible que no es una cabecera, y cada grupo se convierte en una línea separadora.

Ejemplo

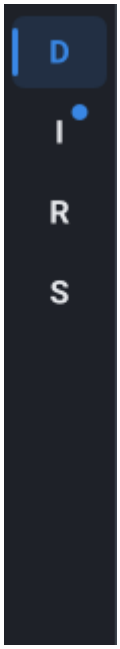
```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupAccordion;
begin
  FNav.Items.BeginUpdate;
  try
    FNav.Items.Clear;
    // a group header owns the items that follow it until the next header
    FNav.Items.AddItem('Sales', '', True);
    FNav.Items.AddItem('Orders', 'O');
    FNav.Items.AddItem('Customers', 'C').Badge := '3';
    FNav.Items.AddItem('Inventory', '', True).Expanded := False;
    FNav.Items.AddItem('Products', 'P');
    FNav.Items.AddItem('Stock', 'S');
    FNav.Items.AddItem('Admin', '', True).Expanded := False;
    FNav.Items.AddItem('Users', 'U');
  finally
    FNav.Items.EndUpdate;
  end;
  FNav.Mode := nbmAccordion;
  // False: expanding a group closes the others
  FNav.AllowMultiExpand := False;
  // expand the Inventory group (index 3) from code
  FNav.Model.ToggleGroup(3);
end;
```

Consulte también

- Modo barra lateral
- TsgcNavigationItems
- Demos

Contraer y animación

Collapsed alterna el ancho de la barra entre **ExpandedWidth** (por defecto 220) y **CollapsedWidth** (por defecto 56), ambos en píxeles lógicos. La barra contraída es un carril de iconos, tanto en modo barra lateral como en modo acordeón:



- Solo se dibuja el hueco del icono de cada elemento. Los títulos se ocultan, y el hint de un elemento sin Hint es su Caption.
- Una cabecera de grupo se convierte en una línea separadora, salvo cuando sería la primera fila o seguiría a otro separador.
- Una insignia se convierte en un punto sobre el icono.
- Con **AnimateCollapse** True (por defecto), el ancho se anima durante unos 150 ms en tiempo de ejecución. En tiempo de diseño, o con AnimateCollapse False, cambia de golpe.
- **OnCollapsedChange** se dispara cuando cambia Collapsed.

La barra establece su propio Width, así que deje Align en alLeft (VCL) o Left (FMX) y deje que el contenido a su derecha use alClient. El modelo informa del ancho con **CurrentWidth**.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupCollapse;
var
  oButton: TButton;
begin
  FNav.ExpandedWidth := 220;
  FNav.CollapsedWidth := 56;
  FNav.AnimateCollapse := True;
  FNav.OnCollapsedChange := OnNavCollapsedChange;

  oButton := TButton.Create(Self);
  oButton.Parent := Self;
  oButton.Caption := 'Collapse';
  oButton.OnClick := OnToggleClick;
end;

procedure TfrmMain.OnToggleClick(Sender: TObject);
begin
  // animates the width to CollapsedWidth, the bar becomes an icon rail
```

```
  FNav.Collapsed := not FNav.Collapsed;
end;

procedure TfrmMain.OnNavCollapsedChange(Sender: TObject);
begin
  if FNav.Collapsed then
    FStatus.Caption := 'Icon rail'
  else
    FStatus.Caption := 'Expanded';
end;
```

Consulte también

- Modo barra lateral
- Insignias
- TsgcVCLNavigationBar

Imágenes y glifos

Cada fila tiene un hueco de icono a su izquierda. Muestra, por este orden de preferencia:

- la imagen **ImageIndex** de la lista de imágenes **Images** de la barra, cuando ambos están establecidos;
- si no, el texto **Glyph** del elemento, centrado, por ejemplo una letra o un símbolo corto;
- si no, nada.

Un glifo no necesita lista de imágenes y sigue los colores del tema, por eso las demos usan letras. Un elemento deshabilitado dibuja su imagen con un 40% de opacidad.

VCL

Images es un TCustomImageList, y las imágenes se dibujan con su propio tamaño. Funcionan tanto las listas de imágenes con máscara como las de 32 bits: cada imagen se convierte una vez a ARGB y se guarda en caché hasta que cambia la lista de imágenes.

FireMonkey

Images es el TCustomImageList de FMX de FMX.ImgList y requiere Delphi XE8 o posterior. Las imágenes se dibujan a 16 x 16 píxeles lógicos, tomadas de la lista de imágenes con el tamaño de dispositivo del fotograma, y se guardan en caché hasta que cambia la lista de imágenes.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupImages;
var
  oImages: TImageList;
  oBitmap: TBitmap;
  i: Integer;
begin
  oImages := TImageList.Create(Self);
  oImages.Width := 16;
  oImages.Height := 16;
  oBitmap := TBitmap.Create;
  try
    oBitmap.Width := 16;
    oBitmap.Height := 16;
    oBitmap.Canvas.Brush.Color := clTeal;
    oBitmap.Canvas.Ellipse(1, 1, 15, 15);
    oImages.AddMasked(oBitmap, clWhite);
  finally
    oBitmap.Free;
  end;

  FNav.Images := oImages;
  // an item with an ImageIndex draws the image, the others their Glyph
  for i := 0 to FNav.Items.Count - 1 do
    if not FNav.Items[i].IsGroup then
      FNav.Items[i].ImageIndex := 0;
end;
```

Consulte también

- TsgcNavigationItem

- Plataformas

Insignias

Badge es un texto corto de un elemento, por ejemplo un contador de no leídos o la palabra New. Se dibuja como una píldora con el color de acento, alineada a la derecha de la fila, con texto negro o blanco, el que se lea mejor sobre el acento. Un Badge vacío no dibuja nada. En el carril de iconos contraído la insignia se convierte en un punto sobre el icono.
Establezca Badge desde código cada vez que cambie el valor, la barra se repinta sola.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupItems;
var
  oItem: TsgcNavigationItem;
begin
  FNav.Items.Clear;
  FNav.Items.AddItem('Workspace', '', True);
  FNav.Items.AddItem('Dashboard', 'D');

  oItem := FNav.Items.Add;
  oItem.Caption := 'Inbox';
  oItem.Glyph := 'I';
  oItem.Badge := '12';
  oItem.Hint := '12 unread messages';

  oItem := FNav.Items.AddItem('Billing', 'B');
  oItem.Enabled := False;
  oItem.Hint := 'Requires an administrator account';

  oItem := FNav.Items.AddItem('Settings', 'S');
  oItem.Tag := 100;
  oItem.OnClick := OnSettingsClick;
end;

procedure TfrmMain.OnSettingsClick(Sender: TObject);
begin
  // Sender is the item
  FStatus.Caption := 'Settings, tag ' +
    IntToStr((Sender as TsgcNavigationItem).Tag);
end;
```

Consulte también

- TsgcNavigationItem
- Temas
- Contraer y animación

Teclado y foco

La barra toma el foco del teclado (**TabStop** en VCL, **CanFocus** en FMX, ambos True por defecto) y mueve una fila con foco por los elementos que pueden recibirlo: los elementos mostrados y habilitados y, en el acordeón expandido, también las cabeceras de grupo.

Tecla	Acción
Arriba, Abajo	Mueve el foco a la fila anterior o siguiente que puede recibirlo y la desplaza hasta hacerla visible.
Inicio, Fin	Mueve el foco a la primera o a la última fila que puede recibirlo.
Enter, Espacio	Activa la fila con foco como un clic: selecciona un elemento y dispara OnItemClick, o alterna un grupo del acordeón.
Izquierda, Derecha	Contrae o expande el grupo del acordeón con foco. Izquierda sobre un elemento de un grupo mueve el foco a su cabecera de grupo.

El rectángulo de foco discontinuo solo se muestra mientras se usa el teclado. La selección y el foco son independientes: **SelectedIndex** es la página que eligió el usuario y **Model.FocusedIndex** la fila donde está el teclado.

Hints

Con **ShowHint** True, el hint es el **Hint** del elemento situado bajo el cursor, o su Caption en el carril contraído, de modo que al pasar a otra fila se muestra su hint. En FMX los hints requieren 10.1 Berlin o posterior.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SelectByCode;
begin
  FNav.TabStop := True;
  FNav.OnSelectionChange := OnNavSelectionChange;
  FNav.OnKeyDown := OnNavKeyDown;
  // select the second item and scroll it into view
  FNav.SelectedIndex := 1;
  FNav.ScrollIntoView(FNav.SelectedIndex);
end;

procedure TfrmMain.OnNavSelectionChange(Sender: TObject);
begin
  if FNav.SelectedIndex >= 0 then
    FStatus.Caption := FNav.Items[FNav.SelectedIndex].Caption;
end;

procedure TfrmMain.OnNavKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  // F2 collapses and expands the bar; the arrows, Home, End, Enter and
  // Space are handled by the bar itself
  if Key = $71 then
    FNav.Collapsed := not FNav.Collapsed;
end;
```

Consulte también

- TsgcVCLNavigationBar
- TsgcNavigationModel
- Modo acordeón

Vínculos de acción

Un elemento se vincula a cualquier TBasicAction, un TAction de VCL o de FireMonkey, mediante su propiedad **Action**. Los vínculos de acción no forman parte de la edición Community, consulte Ediciones.

- **Caption**, **Enabled**, **Visible**, **Hint** e **ImageIndex** del elemento siguen a la acción.
- Un clic en el elemento dispara OnItemClick de la barra y después ejecuta la acción, con la barra como su ActionComponent. Cuando el elemento tiene un manejador OnClick, se ejecuta OnClick en lugar de la acción.
- El OnUpdate de las acciones se ejecuta en reposo, porque la barra llama a InitiateAction en cada elemento.
- En Delphi XE3 y posteriores la acción empuja cada cambio al elemento. En versiones anteriores el elemento copia las propiedades de la acción cuando se asigna y de nuevo en cada InitiateAction.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupActions;
var
  oActions: TActionList;
  oSave: TAction;
  oItem: TsgcNavigationItem;
begin
  oActions := TActionList.Create(Self);
  oSave := TAction.Create(Self);
  oSave.ActionList := oActions;
  oSave.Caption := 'Save';
  oSave.Hint := 'Save the current document';
  oSave.OnExecute := OnSaveExecute;
  oSave.OnUpdate := OnSaveUpdate;

  // Caption, Hint, Enabled, Visible and ImageIndex now follow the action,
  // and a click on the item executes it
  oItem := FNav.Items.Add;
  oItem.Glyph := 'S';
  oItem.Action := oSave;
end;

procedure TfrmMain.OnSaveExecute(Sender: TObject);
begin
  FStatus.Caption := 'Saved';
end;

procedure TfrmMain.OnSaveUpdate(Sender: TObject);
begin
  (Sender as TAction).Enabled := FStatus.Caption <> 'Saved';
end;
```

Consulte también

- TsgcNavigationItem
- Demos
- Ediciones

Temas

Theme establece los colores de la barra a partir de los tokens de tema de sgcUICore: utLight (por defecto), utDark, o utCustom, que usa los colores claros. La superficie, el borde del lado derecho, el texto, las cabeceras de grupo atenuadas, las capas de resaltado y de pulsación y el acento provienen todos del tema.
AccentColor sustituye el acento del tema por un color $AARRGGBB, que usan la barra y el fondo de selección, el rectángulo de foco y las insignias. 0 (por defecto) mantiene el acento del tema.
Los tokens de tema son los mismos que los de sgcDashboard y los demás productos de sgcUICore, así que una barra y un dashboard con el mismo Theme coinciden. Una barra alojada en un tile de sgcDashboard sigue el tema del dashboard, porque el tile empuja su tema a la propiedad publicada Theme de los controles que aloja.

Ejemplo

```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

procedure TfrmMain.SetupTheme;
begin
  FNav.Theme := utDark;
  // $AARRGGBB, 0 uses the accent colour of the theme
  FNav.AccentColor := $FF7C3AED;
end;
```

Consulte también

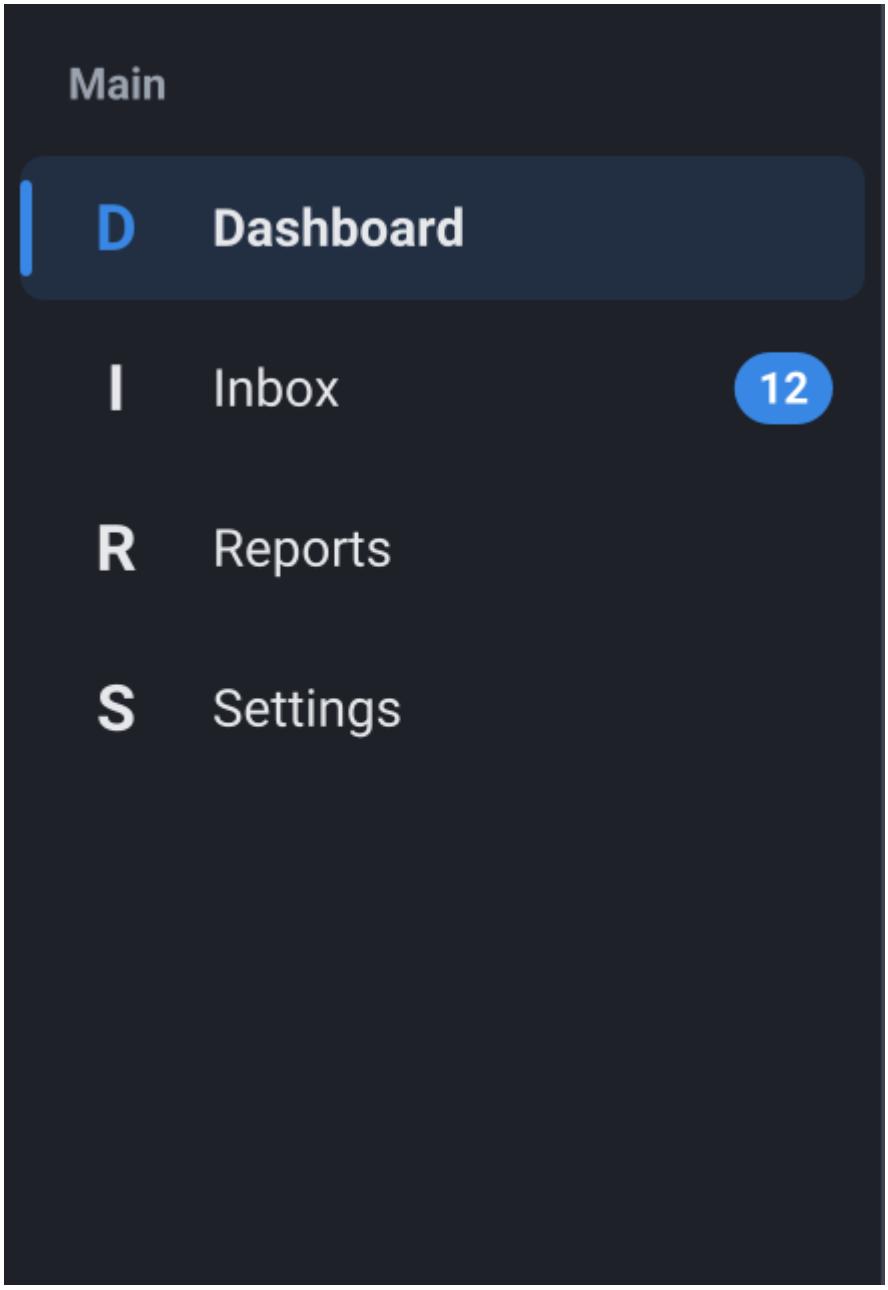
- Arquitectura
- Insignias
- Renderizado sin interfaz

Renderizado sin interfaz

El renderizador de sgcNavigation_Render dibuja un TsgcNavigationModel sin ventana y sin VCL ni FMX, de modo que una aplicación de consola, un servicio, una prueba o un servidor web pueden producir la misma barra como imagen, por ejemplo para la documentación de una aplicación o una prueba visual. La unidad compila en todas las plataformas, Linux64 incluido.

Rutina	Descripción
sgcNavigationSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = nil)	Escribe un archivo o flujo PNG. aWidth y aHeight son píxeles lógicos y aScale los multiplica: 220 por 320 con escala 2 da una imagen de 440 por 640.
sgcNavigationSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = nil)	Escribe un archivo o flujo SVG de la misma barra.
sgcNavigationRender(aModel, aCanvas, aWidth, aHeight, aScale = 1, aScrollY = 0, aImages = nil)	Dibuja la barra en cualquier lienzo de sgcUICore, en píxeles de dispositivo.

aImages es un callback opcional de tipo TsgcNavigationImageEvent que devuelve la imagen de un ImageIndex. Sin él, los elementos dibujan su Glyph. El modelo de un control está disponible mediante su propiedad **Model**, de modo que **sgcNavigationSaveToPNG(Nav.Model, ...)** guarda lo que el control muestra, con cualquier tamaño.



Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_Render;

var
  oModel: TsgcNavigationModel;
  vIndex: Integer;
  vPart: TsgcNavigationPart;
begin
  oModel := TsgcNavigationModel.Create;
  try
    oModel.Items.AddItem('Main', '', True);
    oModel.Items.AddItem('Dashboard', 'D');
    oModel.Items.AddItem('Inbox', 'I').Badge := '12';
    oModel.Items.AddItem('Reports', 'R');
    oModel.Items.AddItem('Settings', 'S');
    oModel.SelectedIndex := 1;
    oModel.Theme := utDark;

    // 220 x 320 logical pixels, written at scale 2 as 440 x 640 pixels
```

```
    sgcNavigationSaveToPNG(oModel, 'navigation.png', 220, 320, 2);
    sgcNavigationSaveToSVG(oModel, 'navigation.svg', 220, 320);

    // the icon rail of the same bar
    oModel.Collapsed := True;
    sgcNavigationSaveToPNG(oModel, 'navigation_rail.png',
      oModel.CurrentWidth, 320);

    // the row under a point, after a Layout at the same width and scale
    oModel.Collapsed := False;
    oModel.Layout(220, 1);
    vIndex := oModel.HitTest(40, 60, vPart);
    if vIndex >= 0 then
      WriteLn('Hit: ', oModel.Items[vIndex].Caption);
    WriteLn('Content height: ', Round(oModel.ContentHeight));
  finally
    oModel.Free;
  end;
end;
```

Consulte también

- TsgcNavigationModel
- Plataformas

Descripción general de .NET

sgcNavigation .NET es un port administrado en C# de la biblioteca Delphi sgcNavigation, no un wrapper alrededor de ella. Las clases C# viven en el espacio de nombres **esegece.sgcNavigation** y conservan los nombres de Delphi, de modo que **TsgcNavigationModel**, **TsgcNavigationItems**, **TsgcNavigationItem** y **TsgcNavigationMode** se leen igual en ambos lenguajes, y la barra de navegación de cada host sigue el patrón de nombres de Delphi. Se dibuja con el port .NET de **sgcUICore**, el mismo lienzo, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, de modo que una barra de navegación se ve igual en WinForms, WPF, Avalonia y en la salida headless en PNG y SVG.
Una barra de navegación muestra una lista de elementos con un icono, un título y una insignia opcional. En el modo sidebar se muestran todos los elementos y las cabeceras de grupo son etiquetas estáticas. En el modo acordeón, un clic en la cabecera de un grupo expande o contrae sus elementos. En ambos modos la barra se pliega en un carril de iconos, animado entre ExpandedWidth y CollapsedWidth.

Paquetes

El port se divide en cinco paquetes NuGet, de modo que un proyecto solo incorpora el host que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcNavigation	El modelo (**TsgcNavigationModel**, **TsgcNavigationItems**, **TsgcNavigationItem**), el vínculo **Command** del elemento, el renderizador **sgcNavigation_Render** y la salida headless en SVG. Sin dependencia nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcNavigation.Rendering	La salida headless en PNG, **sgcNavigation_Rendering**, dibujada sobre el lienzo SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcNavigation.WinForms	**TsgcWinFormsNavigationBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcNavigation.Wpf	**TsgcWpfNavigationBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcNavigation.Avalonia	**TsgcAvaloniaNavigationBar**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0

Los tres paquetes de host dependen de esegece.sgcNavigation.Rendering, que depende de esegece.sgcNavigation, de modo que una línea `PackageReference` trae todo lo que un host necesita. Consulte Instalación desde NuGet.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcNavigation, esegece.sgcNavigation.Rendering, esegece.sgcNavigation.WinForms, esegece.sgcNavigation.Wpf, esegece.sgcNavigation.Avalonia	Todo: las barras de navegación de los tres hosts, los modos sidebar y acordeón, el carril de iconos, imágenes, insignias, navegación con el teclado, el vínculo Command del elemento, los temas y la salida headless en PNG y SVG.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcNavigation.WinForms.Community. Dependen de los paquetes Community de sgcUICore.	El mismo contenido que Registered sin las propiedades **Command** y **CommandParameter** del elemento, ya que la edición Community de Delphi no tiene vínculos de acción. Los elementos se ejecutan mediante su OnClick.

La edición Community es gratuita para uso comercial y no comercial por particulares y organizaciones con menos de 1.000.000 EUR de ingresos brutos anuales y no más de cinco desarrolladores. Una organización que alcance o supere cualquiera de los dos umbrales necesita una licencia Registered. El port .NET no tiene edición Trial.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (TsgcVCLNavigationBar)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	FMX (TsgcFMXNavigationBar)	Windows, .NET Framework 4.7.2 y .NET 8 y 9. La barra se declara en XAML y los elementos se añaden en código.
Avalonia	FMX (TsgcFMXNavigationBar)	Windows, Linux y macOS, .NET 8 y 9, con ratón, táctil y lápiz. La barra se declara en XAML y los elementos se añaden en código.
Headless	sgcNavigation_Render	Cualquier plataforma de los frameworks de destino: SVG con esegece.sgcNavigation, PNG con esegece.sgcNavigation.Rendering.

Todos los hosts tienen la misma API: Items, Mode, Collapsed, ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight, Indent, AllowMultiExpand, SelectedIndex, Theme, AccentColor, Images, AnimateCollapse, ScrollY, ItemAtPos, ItemRect, ScrollIntoView y los eventos OnItemClick, OnSelectionChange y OnCollapsedChange. Consulte Clases de .NET para los nombres de las clases y Diferencias de .NET respecto a Delphi para lo que cambia respecto a Delphi.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Renderizado sin interfaz en .NET
- Demos de .NET

- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, y los paquetes core y de renderizado llegan con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcNavigation.WinForms
dotnet add package esegece.sgcNavigation.Wpf
dotnet add package esegece.sgcNavigation.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcNavigation.WinForms.Community
dotnet add package esegece.sgcNavigation.Wpf.Community
dotnet add package esegece.sgcNavigation.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcNavigation.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie los paquetes Registered o los Community en un proyecto, nunca ambos: contienen los mismos tipos en el mismo espacio de nombres. El código que usa el **Command** del elemento no compila con los paquetes Community; protéjalo con `#if !SGC_COMMUNITY` cuando un proyecto compile ambas ediciones.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto de .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto de .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:sgc="clr-namespace:esegece.sgcNavigation;assembly=esegece.sgcNavigation.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Asigne el espacio de nombres en XAML con `xmlns:sgc="using:esegece.sgcNavigation"`.
- Headless: esegece.sgcNavigation por sí solo basta para SVG. Añada esegece.sgcNavigation.Rendering para PNG, trae SkiaSharp y sus recursos nativos para Windows y Linux.

En C# todo host necesita `using esegece.sgcNavigation;`, y `using esegece.sgcUICore;` para **TsgcUITheme** y los demás tipos de los temas.

Véase también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos de .NET

Inicio rápido WinForms

TsgcWinFormsNavigationBar es la barra de navegación de WinForms, la contrapartida del TsgcVCLNavigationBar de Delphi. Posee un **TsgcNavigationModel**, expone sus propiedades y lo dibuja mediante sgcUICore. El paquete es **esegece.sgcNavigation.WinForms**.

Pasos

- Cree un proyecto WinForms y añada el paquete esegece.sgcNavigation.WinForms.
- Cree un **TsgcWinFormsNavigationBar**. Se acopla a la izquierda por defecto y toma su ancho de **ExpandedWidth**.
- Asigne un **ImageList** a **Images** y añada los elementos con `Items.AddItem(aCaption)`. Defina **ImageIndex**, **Badge** y **Hint** de cada elemento. `AddItem(aCaption, "", true)` añade una cabecera de grupo.
- Asigne **OnSelectionChange** para mostrar la página del elemento seleccionado, y **OnItemClick** para reaccionar a cada clic.
- Ejecute: un clic o Intro selecciona un elemento, las teclas de flecha, Inicio y Fin mueven el foco, y **Collapsed** pliega la barra en el carril de iconos.

Ejemplo

El Program.cs completo de una aplicación WinForms con una sidebar, una insignia, una cabecera de grupo y un botón Collapse. Compila tal cual en un proyecto `net8.0-windows` que referencia esegece.sgcNavigation.WinForms. La barra también está en la paleta de herramientas. El diseñador no serializa los elementos, por lo que siempre se añaden en código, consulte Diferencias de .NET respecto a Delphi.

Véase también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WPF
- Inicio rápido Avalonia
- Imágenes y glifos

Inicio rápido WPF

TsgcWpfNavigationBar es la barra de navegación de WPF. Se declara en XAML como cualquier control, y sus elementos se añaden en código. El paquete es **esegece.sgcNavigation.Wpf**.

Pasos

- Cree un proyecto WPF (`dotnet new wpf`) y añada el paquete esegece.sgcNavigation.Wpf.
- Asigne el espacio de nombres `esegece.sgcNavigation` y declare un **TsgcWpfNavigationBar**. **Mode**, **AnimateCollapse**, **AllowMultiExpand** y las demás propiedades simples se pueden definir como atributos XAML.
- En el code behind asigne **Images**, una lista de **ImageSource**, añada los elementos y asigne los eventos: son delegados, no eventos de WPF, por lo que no se pueden definir como atributos XAML.

Ejemplo

MainWindow.xaml declara la barra, un botón Collapse y un texto que muestra la página seleccionada. El App.xaml del proyecto es el de la plantilla WPF, con StartupUri establecido en MainWindow.xaml.
MainWindow.xaml.cs crea tres iconos vectoriales **DrawingImage**, añade los elementos y asigna los eventos.
Images acepta cualquier **ImageSource**, tanto un **BitmapImage** como un **DrawingImage**. Cuando la lista es una **ObservableCollection<ImageSource>**, la barra se vuelve a dibujar cuando se añade, elimina o reemplaza una imagen.

Véase también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WinForms
- Inicio rápido Avalonia
- Imágenes y glifos

Inicio rápido Avalonia

TsgcAvaloniaNavigationBar es la barra de navegación de Avalonia, la contrapartida del TsgcFMXNavigationBar de Delphi, en Windows, Linux y macOS. Ratón, táctil y lápiz siguen la misma ruta de puntero. El paquete es **esegece.sgcNavigation.Avalonia**.

Pasos

- Cree un proyecto .NET 8 y añada los paquetes esegece.sgcNavigation.Avalonia, Avalonia.Desktop y Avalonia.Themes.Fluent.
- Asigne el espacio de nombres con `xmlns:sgc="using:esegece.sgcNavigation"` y declare un **TsgcAvaloniaNavigationBar** con su **Mode**.
- En el code behind defina el **Theme**, añada los grupos y los elementos, y asigne los eventos: son delegados, no eventos de Avalonia.

Ejemplo

MainWindow.axaml declara un acordeón en el que solo hay un grupo abierto a la vez.
MainWindow.axaml.cs define el tema oscuro, añade tres grupos con sus elementos y muestra el elemento pulsado.
Program.cs inicia la aplicación con el tema Fluent.
Theme oculta la propiedad `StyledElement.Theme` de los controles de Avalonia, por lo que debe definirse en código sobre una variable de tipo TsgcAvaloniaNavigationBar, no como atributo XAML. Images es una lista de **IImage**, un **Bitmap** o un **DrawingImage**.

Véase también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WinForms
- Inicio rápido WPF
- Modo acordeón

Clases de .NET

Cada host tiene una clase de barra de navegación con los mismos miembros. Los tipos neutros respecto al framework son compartidos por los tres hosts y conservan sus nombres de Delphi.

Clases de host

Host	Clase	Clase base	Delphi
WinForms	TsgcWinFormsNavigationBar	TsgcUIWinFormsControl	TsgcVCLNavigationBar
WPF	TsgcWpfNavigationBar	TsgcUIWpfControl	TsgcFMXNavigationBar
Avalonia	TsgcAvaloniaNavigationBar	TsgcUIAvaloniaControl	TsgcFMXNavigationBar

Tipos neutros respecto al framework

Tipo	Descripción
TsgcNavigationModel	Los elementos, el modo, el carril plegado, los elementos seleccionado, bajo el ratón, pulsado y con foco, el diseño de las filas y la prueba de impacto. Cada barra posee uno, consulte TsgcNavigationModel.
TsgcNavigationItems	La colección de elementos: Count, el indexador, Add, AddItem(aCaption, aGlyph = "", aIsGroup = false), Insert, Delete, Clear, IndexOf, BeginUpdate y EndUpdate.
TsgcNavigationItem	Un elemento: Caption, Glyph, ImageIndex, Badge, IsGroup, Expanded, Enabled, Visible, Hint, Tag (long), OnClick, Click() y, en la edición Registered, Command y CommandParameter. Consulte TsgcNavigationItem.
TsgcNavigationMode	nbmSidebar, nbmAccordion.
TsgcNavigationPart	npNone, npItem, npExpander: la parte de una fila que devuelve ItemAtPos.
TsgcNavigationItemEvent	`void (object Sender, TsgcNavigationItem aItem)`, el tipo de OnItemClick.
TsgcUINotifyEvent	`void (object Sender)`, de esegece.sgcUICore.
sgcNavigation_Render, sgcNavigation_Rendering	El renderizador y la salida headless en SVG y PNG, consulte Renderizado sin interfaz en .NET.

Barra de navegación

Miembro	Tipo	Descripción
Model	TsgcNavigationModel	El modelo de la barra.
Items	TsgcNavigationItems	Los elementos y las cabeceras de grupo, en orden.
Mode	TsgcNavigationMode	Sidebar o acordeón.

Collapsed	bool	Pliega la barra en el carril de iconos. El ancho pasa de ExpandedWidth a CollapsedWidth, con animación cuando AnimateCollapse es true.
ExpandedWidth, CollapsedWidth	int	El ancho de la barra expandida y del carril de iconos, 220 y 56 por defecto.
ItemHeight, GroupHeight, Indent	int	La altura de una fila de elemento, de una fila de cabecera de grupo, y la sangría de los elementos de un grupo en el modo acordeón.
AllowMultiExpand	bool	En el modo acordeón, false cierra los demás grupos cuando uno se abre.
SelectedIndex	int	El elemento seleccionado, o -1.
Theme, AccentColor	TsgcUITheme, uint	El tema de la barra. AccentColor 0 usa el color de énfasis del tema.
Images	ImageList, IList<ImageSource>, IList<IImage>	Las imágenes de ImageIndex, consulte Diferencias de .NET respecto a Delphi.
AnimateCollapse	bool	Anima Collapsed durante unos 150 ms.
ScrollY	int	El desplazamiento vertical en píxeles del dispositivo.
ItemAtPos(X, Y), ItemAtPos(X, Y, out aPart)	int	El índice del elemento en un punto del control, o -1, y la parte de la fila.
ItemRect(aIndex)	Rectangle, Rect	La fila de un elemento en las coordenadas del control.
ScrollIntoView(aIndex)	void	Desplaza la barra para que la fila del elemento quede visible.
OnItemClick	TsgcNavigationItemEvent	Un clic, Intro o Espacio en un elemento habilitado, antes de que se ejecute su OnClick o Command. Una cabecera de grupo no lo dispara.
OnSelectionChange	TsgcUINotifyEvent	SelectedIndex cambió, por el usuario o en código.
OnCollapsedChange	TsgcUINotifyEvent	Collapsed cambió.

Las posiciones y los tamaños son píxeles en WinForms (Rectangle) y unidades independientes del dispositivo en WPF y Avalonia (Rect).

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia

Diferencias de .NET respecto a Delphi

La API de .NET sigue la API de Delphi nombre por nombre. Estos son los puntos en los que difiere.

Command en lugar de vínculos de acción

Delphi vincula un elemento a un TBasicAction mediante su propiedad Action, consulte Vínculos de acción. .NET no tiene TBasicAction, por lo que el elemento tiene **Command**, un `System.Windows.Input.ICommand`, y **CommandParameter**. Ambos forman parte solo de la edición Registered, como los vínculos de acción de Delphi.

- `CanExecute(CommandParameter)` controla el **Enabled** del elemento: cuando se asigna el comando o el parámetro y en cada `CanExecuteChanged`.
- Un clic, Intro o Espacio dispara OnItemClick de la barra y después ejecuta el comando, cuando se puede ejecutar. Cuando el elemento tiene un manejador OnClick, se ejecuta OnClick en lugar del comando, como en Delphi.
- Caption, Hint e ImageIndex no se sincronizan desde el comando, porque ICommand no lleva ninguno de ellos. Siguen siendo propiedades del elemento. Visible tampoco se sincroniza.
- No hay OnUpdate en reposo. Un comando que cambia su estado lanza CanExecuteChanged, y el elemento lo sigue de inmediato.

Un programa de consola que vincula un elemento a un comando mínimo, lo ejecuta y lo deshabilita mediante CanExecuteChanged. Solo necesita esegece.sgcNavigation, las barras de los tres hosts se comportan igual.

Images

Delphi usa un TCustomImageList en VCL y un TCustomImageList de FMX.ImgList en FireMonkey, consulte Imágenes y glifos. En .NET la propiedad **Images** tiene el tipo de imagen de cada host, y el **ImageIndex** del elemento es un índice en ella.

Host	Images	Dibujada a
WinForms	System.Windows.Forms.ImageList	El ImageSize de la lista de imágenes, en píxeles.
WPF	IList<ImageSource>: mapas de bits e iconos vectoriales DrawingImage. Una ObservableCollection se vuelve a dibujar cuando cambia.	16 x 16 unidades independientes del dispositivo, nítida con cualquier DPI para los iconos vectoriales.
Avalonia	IList<IImage>: un Bitmap o un DrawingImage. Una lista con INotifyCollectionChanged se vuelve a dibujar cuando cambia.	16 x 16 unidades independientes del dispositivo.

Cada imagen se convierte una vez a ARGB y se mantiene en caché hasta que la lista cambia y, en WPF y Avalonia, también hasta que cambia la escala de DPI. Cuando un elemento no tiene imagen, el texto de su **Glyph** se dibuja en el hueco del icono.

Sugerencias y tooltips

En Delphi la barra muestra el Hint del elemento bajo el cursor cuando ShowHint es True, y FMX muestra sugerencias desde 10.1 Berlin. En .NET la barra siempre muestra un tooltip, sin propiedad ShowHint: el Hint del elemento bajo el cursor, o su Caption en el carril plegado, donde el título no es visible. WinForms usa un componente ToolTip interno, WPF define la propiedad ToolTip de la barra y Avalonia define la propiedad adjunta ToolTip.Tip. No defina usted mismo el tooltip de la barra, la barra lo reemplaza; defina el Hint del elemento.

Unidades

- WinForms: ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight e Indent son píxeles lógicos multiplicados por DPI / 96, como la barra VCL con PPI / 96. ItemAtPos, ItemRect y ScrollY son píxeles del cliente.
- WPF y Avalonia: Width, ExpandedWidth, CollapsedWidth, ItemAtPos e ItemRect son unidades independientes del dispositivo, como en FMX. El marco, el diseño del modelo y **ScrollY** son píxeles del dispositivo, las unidades multiplicadas por la escala de DPI del control (WPF) o la escala de renderizado de la ventana (Avalonia). El BoundsRect de un elemento también está en píxeles del dispositivo.
- Headless: aWidth y aHeight son píxeles lógicos y aScale los multiplica, consulte Renderizado sin interfaz en .NET.

Eventos

OnItemClick, OnSelectionChange, OnCollapsedChange y el OnClick del elemento son propiedades delegado con los nombres de Delphi, no miembros `event` de C#. Asignar un manejador reemplaza el anterior, como un puntero a método de Delphi, y se asignan en código, también en WPF y Avalonia, donde XAML no puede definirlos.

Sin verbos de diseñador

En Delphi la propiedad Items se edita en tiempo de diseño con el editor de colecciones estándar. El port .NET no tiene verbos de diseñador ni editor de colecciones: los Items de una barra no se serializan, por lo que siempre se añaden en código, normalmente en el constructor del formulario o de la ventana. En WinForms la barra está en la paleta de herramientas y sus propiedades simples se definen en la ventana Propiedades. En WPF y Avalonia la barra se declara en XAML.

Ediciones

El port .NET se distribuye como paquetes Registered y Community, consulte Descripción general de .NET. Los paquetes Community no tienen Command ni CommandParameter. No hay edición Trial.

Enlace de datos

En WPF y Avalonia las propiedades visibles para el usuario son propiedades de dependencia y propiedades de Avalonia, por lo que se pueden enlazar, aplicar con estilos y animar, consulte Enlace de datos en .NET.

Véase también

- Descripción general de .NET
- Clases de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Vínculos de acción
- Imágenes y glifos

Enlace de datos en .NET

En los hosts WPF y Avalonia, las propiedades visibles para el usuario de la barra de navegación son propiedades de dependencia (WPF) y propiedades de Avalonia, por lo que se pueden enlazar a un modelo de vista, establecer mediante un Style y animar. Este tema se aplica solo a WPF y Avalonia. El control WinForms no cambia: sus propiedades son propiedades CLR simples, como antes.

Propiedades enlazables

Cada propiedad de la tabla tiene un campo estático con el sufijo Property, por ejemplo `TsgcWpfNavigationBar.SelectedIndexProperty`. En WPF los campos son DependencyProperty. En Avalonia son StyledProperty, excepto SelectedIndex, que es una DirectProperty. Las propiedades marcadas con **TwoWay** se enlazan TwoWay de forma predeterminada, porque la barra también las modifica (SelectedIndex tras un clic o una tecla, ambas tras un cambio en el modelo), de modo que un enlace sin Mode mantiene actualizado el modelo de vista. Las demás propiedades se enlazan OneWay de forma predeterminada.

Clase WPF	Clase Avalonia	Propiedades enlazables
TsgcWpfNavigationBar	TsgcAvaloniaNavigationBar	Mode, **Collapsed (TwoWay)**, ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight, Indent, AllowMultiExpand, **SelectedIndex (TwoWay)**, Theme, AccentColor, AnimateCollapse

En Avalonia, Theme es el tema de sgcUICore (TsgcUITheme) y oculta el tema de control de Avalonia de StyledElement, que sigue accesible mediante `StyledElement.ThemeProperty`.

Comportamiento

- **El modelo conserva los valores.** Cada propiedad se guarda en el modelo de navegación. Un valor enlazado pasa al modelo, y la propiedad recupera el valor que conserva el modelo: el modelo rechaza un ancho o un alto inferior a 1 y mantiene SelectedIndex dentro de los elementos, y la propiedad muestra entonces el valor en uso. Un origen TwoWay también recibe el valor corregido, en WPF y en Avalonia. En WPF llega cuando el dispatcher ha ejecutado el trabajo pendiente. En Avalonia la escritura de vuelta cubre un valor que el origen envía cuando el enlace ya está en marcha. Un valor que el origen ya tiene cuando el enlace se inicia se corrige en el control, pero no se escribe de vuelta en el origen.
- **Los cambios hechos por la barra llegan al modelo de vista.** Un clic en un elemento, una tecla, un cambio en el modelo y un cambio desde el código actualizan SelectedIndex, y un cambio en el modelo o desde el código actualiza Collapsed. Con el enlace TwoWay predeterminado, el modelo de vista sigue.
- **Los eventos se generan para cualquier origen.** OnSelectionChange y OnCollapsedChange se generan ante un cambio por entrada del usuario, por código o por un enlace.
- **Primero los elementos, después el enlace.** SelectedIndex hace referencia a un elemento, así que agregue los elementos antes de que llegue el valor enlazado, por ejemplo estableciendo el DataContext después de los elementos, como hacen los ejemplos.
- **Styles.** Un setter de Style puede establecer cualquier propiedad enlazable, por ejemplo Mode y Theme de todas las barras de navegación de una ventana.

Propiedades que siguen siendo propiedades CLR

Propiedad	Motivo
Items	El árbol de elementos del modelo, construido con AddItem y los demás métodos de la colección, no un valor único que enlazar.
Images	La lista de imágenes de los elementos, un objeto compartido por referencia.

Model	El modelo independiente de la plataforma que hay detrás del control, de solo lectura.
ScrollY	La posición de desplazamiento, que cambia con cada paso de la rueda y no es un ajuste.
OnSelectionChange, OnCollapsedChange y los demás eventos	Propiedades delegado con los nombres de Delphi, asignadas en código, consulte Diferencias de .NET respecto a Delphi.

Ejemplo

El modelo de vista siguiente se comparte en ambos ejemplos. SelectedIndex y Collapsed están enlazados sin Mode, por lo que se enlazan TwoWay: un clic en un elemento cambia el modelo de vista, y la casilla de verificación contrae la barra a través del modelo de vista.

WPF

MainWindow.xaml enlaza la barra. App.xaml es el de la plantilla WPF.
MainWindow.xaml.cs agrega los elementos y después establece el modelo de vista como DataContext.

Avalonia

MainWindow.axaml establece `x:DataType` con el modelo de vista, por lo que los enlaces son enlaces compilados. La clase App es la de Inicio rápido Avalonia.
MainWindow.axaml.cs agrega los elementos y después establece el modelo de vista como DataContext.

Consulte también

- Diferencias de .NET respecto a Delphi
- Clases de .NET
- Inicio rápido WPF
- Inicio rápido Avalonia

Renderizado sin interfaz en .NET

Una barra de navegación se puede dibujar sin ventana, en un servidor, en una prueba o en un informe. **sgcNavigation_Render.sgcNavigationSaveToSVG** escribe un archivo o flujo SVG solo con el paquete esegece.sgcNavigation, y **sgcNavigation_Rendering.sgcNavigationSaveToPNG** escribe un PNG con esegece.sgcNavigation.Rendering, que dibuja sobre SkiaSharp en Windows, Linux y macOS. La salida SVG es idéntica byte a byte a la del renderizado sin interfaz de Delphi.

Métodos

Método	Paquete	Descripción
sgcNavigation_Render.sgcNavigationSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = null)	esegece.sgcNavigation	Escribe la barra de aModel como SVG.
sgcNavigation_Rendering.sgcNavigationSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = null)	esegece.sgcNavigation.Rendering	Escribe la barra de aModel como PNG sobre un fondo transparente.
sgcNavigation_Render.sgcNavigationRender(aModel, aCanvas, aWidth, aHeight, aScale = 1, aScrollY = 0, aImages = null)	esegece.sgcNavigation	Dibuja la barra en cualquier lienzo de sgcUICore.

aWidth y aHeight son píxeles lógicos. La imagen mide aWidth x aScale por aHeight x aScale píxeles, de modo que una escala de 2 da una imagen nítida para una pantalla de alta densidad o para imprimir. aImages es un **TsgcNavigationImageEvent** opcional, `bool (int aImageIndex, out TsgcUIARGBBuffer aImage)`, que devuelve la imagen de un ImageIndex como un **TsgcUIARGBBuffer**; sin él se dibuja el Glyph del elemento en el hueco del icono.

Ejemplo

Un programa de consola (`net8.0`) que referencia esegece.sgcNavigation.Rendering y escribe en PNG y SVG una sidebar con una insignia, una cabecera de grupo y un elemento deshabilitado, el carril de iconos plegado, y los mismos elementos como un acordeón oscuro con un grupo cerrado.

Véase también

- Descripción general de .NET
- Clases de .NET
- Renderizado sin interfaz
- TsgcNavigationModel

Demos de .NET

Las demos de .NET están en la carpeta `demos` del producto y en el zip de demos, con la solución `sgcNavigation_Demos.sln`. Cada demo tiene un README.md.

Demo	Host	Descripción
SidebarApp	WinForms	Una sidebar a la izquierda cambia las páginas de una aplicación. Los cinco elementos (Dashboard, Inbox, Reports, Billing y Settings) toman sus iconos de un ImageList, Inbox y Reports muestran una insignia y Billing está deshabilitado. El botón Collapse pliega la barra en el carril de iconos con una animación. OnSelectionChange muestra la página guardada en el Tag del elemento seleccionado, OnItemClick escribe el elemento pulsado en la línea de estado y OnCollapsedChange actualiza el título del botón. La primera página aloja un dashboard sgcDashboard con dos tarjetas KPI en vivo, y el combo Theme alterna la barra, el dashboard y las páginas entre claro y oscuro.
SidebarAppWpf	WPF	La misma aplicación sobre un TsgcWpfNavigationBar con iconos vectoriales DrawingImage y un TsgcWpfDashboard declarado en XAML.
SidebarAppAvalonia	Avalonia	La misma aplicación sobre un TsgcAvaloniaNavigationBar con iconos vectoriales IImage, en Windows, Linux y macOS.
Accordion	WinForms	Un acordeón con los grupos Sales, Inventory y Admin. Con "Allow multi expand" sin marcar, abrir un grupo cierra los demás. New order, Export y Audit log están vinculados a comandos mediante el Command del elemento: un clic ejecuta el comando y el elemento Export sigue a CanExecute, que cambia la casilla "Export command enabled". El botón Rename renombra el propio elemento, porque ICommand no lleva título. En la edición Community los elementos se ejecutan mediante su OnClick.
AccordionWpf	WPF	El mismo acordeón sobre un TsgcWpfNavigationBar.
AccordionAvalonia	Avalonia	El mismo acordeón sobre un TsgcAvaloniaNavigationBar.

Compilar y ejecutar

Las demos necesitan el SDK de .NET 8 o posterior. Ejecute una demo desde su carpeta:

```
dotnet run
```

Dentro del árbol del producto las demos se compilan contra los proyectos de la biblioteca. Desde el zip de demos restauran los paquetes NuGet, la edición Community por defecto. Defina **SgcNavigationEdition** para usar los paquetes Registered:

```
dotnet build -p:SgcNavigationEdition=Registered
```

Autocomprobación

Cada demo tiene un modificador `--test` que abre la ventana fuera de pantalla, pulsa los elementos, abre y cierra los grupos, pliega la barra y cambia el tema, escribe una línea PASS o FAIL por comprobación en el archivo y termina con el número de comprobaciones fallidas, de modo que 0 significa que todas pasaron.

```
dotnet run -- --test result.txt
```

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos

Solución de problemas

Undeclared identifier Action, o "Property Action does not exist"

La edición Community compila sin SGC_NAVIGATION_ACTIONS, por lo que la propiedad Action de los elementos no existe. El código que la usa falla con E2003 Undeclared identifier, y un formulario guardado por una compilación Registered con un elemento vinculado a una acción provoca una excepción "Error reading" al cargarse. Use el OnClick del elemento o el evento OnItemClick de la barra, o una compilación Registered o Trial. Consulte Ediciones.

Un clic en una cabecera de grupo no la selecciona

Las cabeceras de grupo nunca se seleccionan. En modo barra lateral son etiquetas estáticas, y en el acordeón expandido un clic alterna el grupo. Use un elemento con IsGroup False para una página. Consulte Modo acordeón.

Un elemento no reacciona a un clic

Un elemento deshabilitado (Enabled False) se dibuja atenuado e ignora los clics y el teclado. Un elemento con una Action sigue el Enabled de la acción, así que compruebe también el OnUpdate de la acción.

Las imágenes no se muestran en FireMonkey

La propiedad Images de la barra FMX requiere Delphi XE8 o posterior. En versiones anteriores los elementos dibujan su Glyph. Consulte Imágenes y glifos.

No hay hints en la barra FMX

Los hints de los elementos de la barra FMX requieren Delphi 10.1 Berlin o posterior, y ShowHint debe ser True.

La barra salta a su nuevo ancho en lugar de animarse

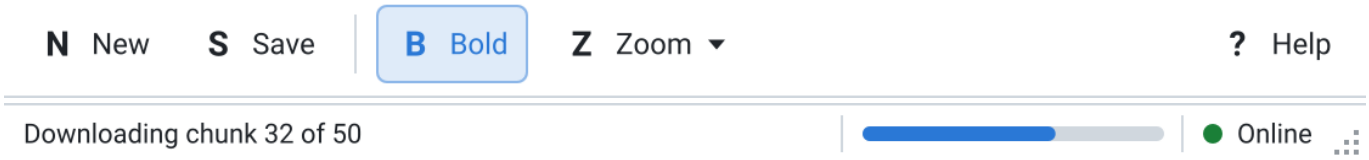
La animación de contracción solo se ejecuta en tiempo de ejecución y solo mientras AnimateCollapse es True. En tiempo de diseño el ancho cambia de golpe.

La demo SidebarApp, SidebarFMX o Accordion no compila

SidebarApp y SidebarFMX alojan un dashboard y requieren sgcDashboard instalado. Accordion usa vínculos de acción, que no forman parte de la edición Community. Consulte Demos.

Introducción

sgcToolbar es una barra de herramientas y una barra de estado nativas para Delphi y C++Builder, para VCL y FireMonkey. **TsgcVCLToolbar** y **TsgcFMXToolbar** muestran una fila de botones, interruptores, grupos de opciones, botones desplegables, etiquetas, separadores y espaciadores, y mueven los elementos que no caben a un menú de desbordamiento. **TsgcVCLStatusBar** y **TsgcFMXStatusBar** muestran paneles de texto, barras de progreso e indicadores de estado de colores, y un hilo de trabajo puede actualizarlos directamente con setters seguros entre hilos.
sgcToolbar forma parte de la suite visual de eSeGeCe y da su marco a los demás productos: una barra de herramientas arriba y una barra de estado abajo en una ventana que aloja un panel **sgcDashboard**, una cuadrícula **sgcGrid** o una barra de navegación **sgcNavigation**. Se dibuja con **sgcUICore**, el paquete de ejecución compartido con el lienzo, el rasterizador, las fuentes, el escritor de SVG y los temas claro y oscuro, de modo que las barras se ven igual en VCL, en FireMonkey y en la salida PNG y SVG sin interfaz. sgcToolbar requiere sgcUICore, que se instala junto con él.



Ediciones

sgcToolbar se distribuye como Registered, Trial y Community. La edición Community incluye todas las características excepto el menú de desbordamiento y los vínculos de acción de los elementos. Consulte Ediciones.

Por dónde empezar

- Arquitectura: los modelos independientes del framework, el renderizador sobre el lienzo de sgcUICore y los controles VCL y FMX.
- Plataformas: las versiones de Delphi, C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX e Inicio rápido C++Builder: una primera barra de herramientas y barra de estado en pocos minutos.
- Componentes: las barras de herramientas, las barras de estado, los elementos, los paneles y los modelos.
- Tipos de elemento, Interruptores y grupos de opciones, Menús desplegables, Desbordamiento, Vínculos de acción, Imágenes y glifos y Teclado.
- Paneles de estado, Progreso e indicadores, Actualizaciones seguras entre hilos y Tirador de tamaño.
- Temas y Renderizado sin interfaz.
- Demos: los proyectos de demostración de Delphi y C++Builder.

Arquitectura

sgcToolbar está construido en capas. Cada capa solo depende de las inferiores, por lo que los modelos y el renderizador compilan sin VCL ni FireMonkey.

Capa	Unidad	Función
Modelos	sgcToolbar_Model	TsgcToolbarModel contiene los Items (una colección de TsgcToolbarItem), ShowCaptions, los tamaños, el tema y el estado de interacción (elemento resaltado, pulsado y con foco). TsgcStatusBarModel contiene los Panels (una colección de TsgcStatusBarPanel), el tema y el tirador de tamaño. Sus métodos Layout calculan el rectángulo de cada elemento o panel, y HitTest encuentra el que hay bajo un punto.
Renderizador	sgcToolbar_Render	RenderToolbar y RenderStatusBar pintan un modelo en cualquier lienzo de sgcUICore. Los métodos SaveToPNG y SaveToSVG de los modelos los usan para escribir una imagen sin ventana. Consulte Renderizado sin interfaz.
Controles VCL	sgcToolbar_VCL	TsgcVCLToolbar y TsgcVCLStatusBar poseen un modelo, publican sus propiedades y lo pintan mediante la base de controles VCL de sgcUICore. Añaden el ratón, el teclado, los hints, la lista de imágenes, los menús emergentes y el tirador de tamaño.
Controles FMX	sgcToolbar_FMX	TsgcFMXToolbar y TsgcFMXStatusBar son los controles FireMonkey con las mismas propiedades, métodos y eventos, además de entrada táctil.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme	El runtime compartido: el lienzo vectorial y el rasterizador, la base de controles VCL y FMX con control de fotogramas (MaxFPS) y los tokens de los temas claro y oscuro.

Modelo y control

Las propiedades publicadas de una barra son las propiedades de su modelo, y la propiedad **Model** de la barra da acceso al resto: Layout, HitTest, la lista de desbordamiento y el estado de interacción. Un cambio del modelo programa un nuevo fotograma y nunca pinta directamente, de modo que muchos cambios seguidos cuestan un solo repintado.

Pintado

El renderizador pinta cada elemento y cada panel: el hueco del icono con una imagen o un glifo, el título, la flecha del desplegable, los estados resaltado, pulsado, hundido, con foco y deshabilitado, los separadores, el botón de desbordamiento, las barras de progreso, los puntos indicadores y el tirador de tamaño. Todos los tamaños son píxeles lógicos multiplicados por la escala del monitor, por lo que las barras se ven nítidas con cualquier DPI. Los controles VCL usan PPI / 96 como escala y los controles FMX la escala del formulario.

Altura

La altura de una barra de herramientas sigue a sus botones: ButtonSize más dos veces Padding, en píxeles lógicos, y cualquier otra altura se ignora. Una barra de estado conserva la Height que se le da, 24 píxeles lógicos por defecto.

Consulte también

- Plataformas
- TsgcToolbarModel
- TsgcStatusBarModel
- Temas

Ediciones

sgcToolbar se compila desde el mismo código fuente para todas las ediciones. La edición solo cambia las características que incluye una compilación.

Edición	Incluye
Registered	Todo: las barras de herramientas y de estado VCL y FMX, todos los tipos de elemento, interruptores y grupos de opciones, menús desplegables, el menú de desbordamiento, los vínculos de acción de los elementos, imágenes y glifos, la navegación con teclado, los paneles de texto, progreso e indicador, los setters seguros entre hilos, el tirador de tamaño, los temas y la salida PNG y SVG sin interfaz.
Trial	Las mismas características que Registered. Una compilación con SGC_TRIAL muestra un aviso de versión de prueba cuando pasa la ventana de lanzamiento indicada en la compilación. Nada más cambia.
Community	Todo excepto el menú de desbordamiento y los vínculos de acción de los elementos.

Qué omite Community

La edición Community compila con SGC_TOOLBAR_ADVANCED sin definir. En una compilación Community:

- La barra de herramientas no tiene botón de desbordamiento. Los elementos que no caben se recortan en el borde derecho, y la propiedad **OverflowMenu** de las barras de herramientas no existe. Consulte Desbordamiento.
- The **Action** de TsgcToolbarItem no existe, por lo que un elemento no se puede vincular a un TAction. Asigne en su lugar el **OnClick** del elemento o el evento **OnItemClick** de la barra de herramientas. Consulte Vínculos de acción.

La edición es una decisión en tiempo de compilación en sgcVer.inc (SGC_COMMUNITY, que también desactiva SGC_TRIAL), no un interruptor en tiempo de ejecución. El código o el archivo de formulario que usa Action u OverflowMenu no compila ni se carga con una compilación Community, consulte Solución de problemas.

Consulte también

- Desbordamiento
- Vínculos de acción
- Plataformas

Instalación

sgcToolbar forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcToolbar** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcToolbar

sgcUICore, la base compartida de los productos de disposición y de sgcCharts, se incluye con sgcToolbar. No tiene casilla propia. El instalador la instala automáticamente siempre que se selecciona sgcToolbar. sgcToolbar no tiene paquete de puente de sgcWebSockets.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Toolbar`	Las demos de sgcToolbar. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluido sgcToolbar.
- **Community**: el instalador gratuito Community de sgcUI incluye sgcToolbar. Sus productos son sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard, mientras que sgcPDF y sgcMaps no forman parte de Community. Consulte Ediciones para saber qué deja fuera la compilación Community de sgcToolbar.

Instalaciones independientes anteriores

Si el instalador encuentra una instalación independiente anterior de sgcUICore, como la que escribió el instalador independiente de sgcCharts, muestra las rutas de bibliotecas y los paquetes de diseño de esa instalación en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcToolbarD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcToolbar<V>` / `dclsgcToolbar<V>`	Los componentes de sgcToolbar (tiempo de ejecución / tiempo de diseño).

Compílelos en orden de dependencia: `sgcUICore`, después `sgcToolbar`, después `dclsgcToolbar`. Después instale el paquete de diseño `dclsgcToolbar` y añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` correspondiente a su versión, por ejemplo `sgcToolbarC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

.NET

No cambia nada para .NET. sgcToolbar se sigue distribuyendo como sus propios paquetes NuGet (`esegece.sgcToolbar` y sus paquetes complementarios). Consulte la instalación de .NET.

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, de XE a XE8, de 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, de XE a XE8, de 10 a 10.4, 11, 12 y 13. Los paquetes generan los encabezados .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	TsgcVCLToolbar y TsgcVCLStatusBar en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	TsgcFMXToolbar y TsgcFMXStatusBar a partir de Delphi XE6 (SGC_TOOLBAR_FMX), en Win32, Win64, macOS, iOS y Android, y en Linux con FMX for Linux (FMXLinux). La propiedad Images necesita XE8 o posterior (FMX.ImgList), y los hints necesitan 10.1 Berlin o posterior.
Linux64	Las unidades independientes del framework, los modelos y el renderizador con la salida PNG y SVG sin interfaz, a partir de Delphi 10.2 (SGC_TOOLBAR_LINUX).
.NET	Un port a C# con la misma API: TsgcWinFormsToolbar y TsgcWinFormsStatusBar, TsgcWpfToolbar y TsgcWpfStatusBar, TsgcAvaloniaToolbar y TsgcAvaloniaStatusBar, y la salida sin interfaz en PNG y SVG, en .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Paquetes

sgcToolbar incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcToolbarD13.dpk para Delphi 13 y sgcToolbarC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcToolbar	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, sgcUICore
sgcToolbarVCL	Runtime, VCL	sgcToolbar_VCL (TsgcVCLToolbar, TsgcVCLStatusBar)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcToolbar
sgcToolbarFMX	Runtime, FireMonkey, desde XE6	sgcToolbar_FMX (TsgcFMXToolbar, TsgcFMXStatusBar)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcToolbar
dclsgcToolbar	Tiempo de diseño, VCL	Registra TsgcVCLToolbar y TsgcVCLStatusBar en la página de paleta **SGC Toolbar**.	designide, rtl, vcl, sgcToolbar, sgcToolbarVCL
dclsgcToolbarFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXToolbar y TsgcFMXStatusBar en la página de paleta **SGC Toolbar**.	designide, rtl, fmx, sgcToolbar, sgcToolbarFMX, dclsgcToolbar

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcToolbarFMX requiere dclsgcToolbar, así que instale primero dclsgcToolbar.

Versiones del IDE y plataformas de destino

Delphi	sgcToolbar	sgcToolbarVCL	sgcToolbarFMX	dclsgcToolbar, dclsgcToolbarFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcToolbar, sgcToolbarVCL	sgcToolbarFMX	dclsgcToolbar, dclsgcToolbarFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcToolbar)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcToolbar)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcToolbarC13.cbproj, sgcToolbarVCLC13.cbproj, sgcToolbarFMXC13.cbproj, dclsgcToolbarC13.cbproj y dclsgcToolbarFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXToolbar y TsgcFMXStatusBar) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcToolbarVCL solo funcionan en Windows.

- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcToolbar_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcToolbar se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa las unidades independientes del framework. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.
Las barras VCL solo aparecen en la paleta de un formulario VCL y las barras FMX solo en la paleta de un formulario FireMonkey. Las propiedades **Items** y **Panels** se editan con el editor de colecciones estándar.

Consulte también

- Ediciones
- Arquitectura
- Solución de problemas

Inicio rápido VCL

En tiempo de diseño

- Coloque un **TsgcVCLToolbar** de la página de paleta **SGC Toolbar** en un formulario. Se alinea arriba por defecto.
- Haga doble clic en **Items** en el Object Inspector y añada los elementos. Establezca **Caption** y **Glyph** (una letra o un símbolo corto) de cada botón, y establezca **Kind** en tikSeparator, tikToggle, tikDropdown, tikLabel o tikSpacer donde necesite uno.
- Coloque un **TsgcVCLStatusBar**. Se alinea abajo por defecto. Haga doble clic en **Panels** y añada los paneles, con **Spring** en True en el que ocupa el ancho libre.
- Escriba un manejador **OnItemClick** para la barra de herramientas, o asigne el **OnClick** de cada elemento.
- Establezca **Theme** en utDark en ambas barras para usar la paleta oscura.

En código

Las mismas barras creadas en tiempo de ejecución. El programa compila tal cual con `sgcUI\Source` en la ruta de búsqueda.

```
uses
  Forms, Classes, Controls, StdCtrls,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

procedure TMainForm.CreateBars;
var
  oItem: TsgcToolbarItem;
  oPanel: TsgcStatusBarPanel;
begin
  // the toolbar, aligned to the top of the form by default
  Toolbar := TsgcVCLToolbar.Create(Self);
  Toolbar.Parent := Self;
  Toolbar.ShowHint := True;
  Toolbar.OnItemClick := ToolbarItemClick;

  oItem := Toolbar.Items.Add;
  oItem.Caption := 'New';
  oItem.Glyph := 'N';
  oItem.Hint := 'Create a new document';
  oItem := Toolbar.Items.Add;
  oItem.Caption := 'Save';
  oItem.Glyph := 'S';
  Toolbar.Items.Add.Kind := tikSeparator;
  oItem := Toolbar.Items.Add;
  oItem.Kind := tikToggle;
  oItem.Caption := 'Bold';
  oItem.Glyph := 'B';
  // a spacer takes the free width and pushes the next items to the right
  Toolbar.Items.Add.Kind := tikSpacer;
  oItem := Toolbar.Items.Add;
  oItem.Caption := 'Help';
  oItem.Glyph := '?';

  // the status bar, aligned to the bottom of the form by default
  StatusBar := TsgcVCLStatusBar.Create(Self);
  StatusBar.Parent := Self;
  oPanel := StatusBar.Panels.Add;
  oPanel.Spring := True;
  oPanel.Text := 'Ready';
  oPanel := StatusBar.Panels.Add;
  oPanel.Kind := spkIndicator;
  oPanel.AutoSize := True;
```

```
  oPanel.Text := 'Saved';
  oPanel.IndicatorState := sisOk;

  Memo := TMemo.Create(Self);
  Memo.Parent := Self;
  Memo.Align := alClient;
end;

procedure TMainForm.ToolbarItemClick(Sender: TObject; aItem: TsgcToolbarItem);
begin
  StatusBar.SetPanelText(0, aItem.Caption + ' clicked');
end;
```

Consulte también

- TsgcVCLToolbar
- TsgcVCLStatusBar
- TsgcToolbarItem
- Tipos de elemento
- Paneles de estado

Inicio rápido FMX

Los controles FireMonkey necesitan Delphi XE6 o posterior. Se usan exactamente igual que los VCL: coloque un **TsgcFMXToolbar**, que se alinea arriba por defecto, y un **TsgcFMXStatusBar**, alineado abajo, rellene **Items** y **Panels** con el editor de colecciones y gestione **OnItemClick**. Un toque en un botón funciona como un clic.

En código

```
uses
  System.Classes, FMX.Forms, FMX.Types, FMX.Memo,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_FMX;

procedure TMainForm.CreateBars;
var
  oItem: TsgcToolbarItem;
  oPanel: TsgcStatusBarPanel;
begin
  // the toolbar, aligned to the top by default; its height follows ButtonSize
  Toolbar := TsgcFMXToolbar.Create(Self);
  Toolbar.Parent := Self;
  Toolbar.Theme := utDark;
  Toolbar.OnItemClick := ToolbarItemClick;
  oItem := Toolbar.Items.Add;
  oItem.Caption := 'Open';
  oItem.Glyph := 'O';
  oItem := Toolbar.Items.Add;
  oItem.Kind := tikToggle;
  oItem.Caption := 'Wrap';
  oItem.Glyph := 'W';

  // the status bar, aligned to the bottom by default
  StatusBar := TsgcFMXStatusBar.Create(Self);
  StatusBar.Parent := Self;
  StatusBar.Theme := utDark;
  oPanel := StatusBar.Panels.Add;
  oPanel.Spring := True;
  oPanel.Text := 'Ready';
  oPanel := StatusBar.Panels.Add;
  oPanel.Kind := spkProgress;
  oPanel.Width := 120;

  Memo := TMemo.Create(Self);
  Memo.Parent := Self;
  Memo.Align := TAlignLayout.Client;
end;

procedure TMainForm.ToolbarItemClick(Sender: TObject; aItem: TsgcToolbarItem);
begin
  if aItem.Kind = tikToggle then
    Memo.WordWrap := aItem.Down
  else
    StatusBar.SetPanelText(0, aItem.Caption + ' clicked');
end;
```

Consulte también

- TsgcFMXToolbar
- TsgcFMXStatusBar
- Plataformas

Inicio rápido C++Builder

Instale los paquetes de sgcToolbar para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de inclusión y de bibliotecas del proyecto, y use los componentes de la página de paleta **SGC Toolbar** exactamente igual que en Delphi. El instalador añade ambas rutas al IDE por usted. Los encabezados .hpp y los archivos .lib generados de sgcToolbar y sgcUICore están en esa única carpeta, por lo que sgcUICore no necesita una ruta propia.

En código

OnItemClick es un evento de tipo closure, por lo que su manejador es una función miembro __fastcall de una clase.

Consulte también

- Inicio rápido VCL
- Demos
- Plataformas

Demos

Las demos están en la carpeta **Demos** de la instalación, con los proyectos de Delphi en Demos\Delphi y las versiones para C++Builder de las demos VCL en Demos\CBuilder. Cada carpeta de demo tiene un readme.html que explica qué hace la demo y cómo funciona.

Demo	Componentes	Qué muestra
EditorShell	TsgcVCLToolbar, TsgcVCLStatusBar, TActionList	La estructura de un pequeño editor de texto: botones New, Open y Save vinculados a acciones, los interruptores Bold e Italic, un grupo de opciones Left, Center y Right, un desplegable Zoom, un espaciador, un interruptor de tema Dark y un menú de desbordamiento, y una barra de estado con la posición del cursor y un indicador de estado guardado. Delphi y C++Builder, Win32 y Win64. Los vínculos de acción y el menú de desbordamiento no forman parte de la edición Community, por lo que esta demo no compila con ella.
EditorShellFMX	TsgcFMXToolbar, TsgcFMXStatusBar, TActionList	La demo EditorShell para FireMonkey, con los mismos elementos, acciones y paneles. Delphi, Win32 y Win64. También necesita una compilación Registered o Trial.
LiveStatus	TsgcVCLStatusBar, TsgcVCLToolbar, TThread	Un hilo de trabajo simula una descarga y actualiza un panel de texto elástico, un panel de progreso, un panel indicador y un contador AutoSize directamente con SetPanelText, SetProgress y SetIndicator, sin Synchronize. Delphi y C++Builder, Win32 y Win64.

Consulte también

- Inicio rápido VCL
- Actualizaciones seguras entre hilos
- Vínculos de acción

TsgcVCLToolbar

Unidad: sgcToolbar_VCL
TsgcVCLToolbar es la barra de herramientas VCL, solo para Windows. Se alinea arriba por defecto y muestra sus Items de izquierda a derecha: consulte Tipos de elemento. Gestiona el ratón, el teclado, los hints de los elementos, los menús desplegables y el menú de desbordamiento. Su altura sigue a ButtonSize y Padding, multiplicados por PPI / 96.

Propiedades

Propiedad	Tipo	Descripción
Items	TsgcToolbarItems	Los elementos, en orden de visualización. Consulte TsgcToolbarItems.
ShowCaptions	Boolean	Dibuja el título junto al icono. Cuando es False, los botones con icono muestran solo el icono y el título pasa a ser el hint. Un elemento sin Glyph ni ImageIndex siempre muestra su título. Por defecto True.
ButtonSize	Integer	Altura de un botón, y ancho de un botón sin título, en píxeles lógicos. Por defecto 32.
Spacing	Integer	Separación entre dos elementos en píxeles lógicos. Por defecto 4.
Padding	Integer	Espacio alrededor de los botones en píxeles lógicos. La altura de la barra es ButtonSize más dos veces Padding. Por defecto 6.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom, que usa los colores claros. Consulte Temas.
AccentColor	TsgcUIColor	Color del anillo de foco y de un interruptor pulsado, como $AARRGGBB. 0 (por defecto) usa el acento del tema.
Images	TCustomImageList	La lista de imágenes del ImageIndex de los elementos. Cada imagen se convierte una sola vez a ARGB directo y se guarda en caché hasta que la lista cambia. Consulte Imágenes y glifos.
Model	TsgcToolbarModel (public, solo lectura)	El modelo de la barra. Consulte TsgcToolbarModel.
OverflowMenu	TPopupMenu (public, solo lectura)	El menú del botón de desbordamiento, reconstruido cada vez que se muestra. No está en la edición Community. Consulte Desbordamiento.
MaxFPS	Integer	El límite de control de fotogramas de la base de controles de sgcUICore.

Los miembros habituales de un control también están publicados: **Align** (por defecto alTop), **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop** (por defecto False), **Visible**, OnClick, OnContextPopup, OnEnter, OnExit, OnMouseDown, OnMouseMove, OnMouseUp y OnResize.

Métodos

Método	De-vuelve	Descripción
ItemAtPos(X, Y: Integer)	Integer	El índice del elemento en un punto del control, CS_TOOLBAR_HIT_OVERFLOW (-2) para el botón de desbordamiento o CS_TOOLBAR_HIT_NONE (-1). Los separadores y los espaciadores nunca reciben el impacto.
ItemRect(aIndex: Integer)	TRect	El rectángulo de un elemento en coordenadas del control, o un rectángulo vacío cuando el elemento no está en la barra.
OverflowRect	TRect	El rectángulo del botón de desbordamiento, vacío cuando no hay desborda-miento.
ClickItem(aIndex: Integer)		Activa un elemento como lo haría un clic: invierte un interruptor, dispara OnItemClick, ejecuta el OnClick del elemento o su Action y muestra el menú de un desplegable.
InitiateAction		Permite que cada elemento con una Action se actualice. El framework lo lla-ma en reposo.

Eventos

Evento	Tipo	Descripción
OnItem-Click	TsgcTool-barItemE-vent	Se hizo clic, se tocó o se eligió en el menú de desbordamiento un botón, interruptor o desplegable habilitado, o se activó con Enter o Espacio. Down de un interruptor ya ha cambiado. El OnClick del elemento o su Action se ejecuta después de este evento.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatus-Bar) y un **Memo** (TMemo).

```
var
  vIndex: Integer;
  vRect: TRect;
  vPoint: TPoint;
begin
  Toolbar.ShowCaptions := False;       // icons only; the caption becomes the hint
  Toolbar.ButtonSize := 28;            // the height follows: 28 + 2 x Padding
  vPoint := Toolbar.ScreenToClient(Mouse.CursorPos);
  vIndex := Toolbar.ItemAtPos(vPoint.X, vPoint.Y);
  if vIndex >= 0 then
  begin
    vRect := Toolbar.ItemRect(vIndex);
    Caption := Format('%s at x=%d', [Toolbar.Items[vIndex].Caption, vRect.Left]);
  end
  else if vIndex = CS_TOOLBAR_HIT_OVERFLOW then
    Caption := 'The overflow button';
end;
```

Consulte también

- TsgcFMXToolbar

- TsgcToolbarItem
- TsgcToolbarModel
- Inicio rápido VCL

TsgcFMXToolbar

Unidad: sgcToolbar_FMX (Delphi XE6 o posterior)
TsgcFMXToolbar es la barra de herramientas de FireMonkey con las mismas propiedades, métodos y eventos que TsgcVCLToolbar. Se alinea arriba por defecto y su altura es ButtonSize más dos veces Padding, en unidades lógicas. Gestiona el ratón y el tacto, el teclado cuando puede recibir el foco, y muestra el TPopupMenu de FireMonkey de un elemento desplegable y del botón de desbordamiento.

Propiedades

Propiedad	Tipo	Descripción
Items	TsgcToolbarItems	Los elementos, en orden de visualización. Consulte TsgcToolbarItems.
ShowCaptions	Boolean	Dibuja el título junto al icono. Cuando es False, los botones con icono muestran solo el icono y el título pasa a ser el hint. Un elemento sin Glyph ni ImageIndex siempre muestra su título. Por defecto True.
ButtonSize	Integer	Altura de un botón, y ancho de un botón sin título, en píxeles lógicos. Por defecto 32.
Spacing	Integer	Separación entre dos elementos en píxeles lógicos. Por defecto 4.
Padding	Integer	Espacio alrededor de los botones en píxeles lógicos. La altura de la barra es ButtonSize más dos veces Padding. Por defecto 6.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom, que usa los colores claros. Consulte Temas.
AccentColor	TsgcUIColor	Color del anillo de foco y de un interruptor pulsado, como $AARRGGBB. 0 (por defecto) usa el acento del tema.
Images	TCustomImageList	La lista de imágenes del ImageIndex de los elementos. Necesita XE8 o posterior, y las imágenes se dibujan a 16 x 16 píxeles lógicos. Consulte Imágenes y glifos.
Model	TsgcToolbarModel (public, solo lectura)	El modelo de la barra. Consulte TsgcToolbarModel.
OverflowMenu	TPopupMenu (public, solo lectura)	El menú del botón de desbordamiento, reconstruido cada vez que se muestra. No está en la edición Community. Consulte Desbordamiento.
MaxFPS	Integer	El límite de control de fotogramas de la base de controles de sgcUICore.

Los miembros habituales de un control FireMonkey también están publicados: **Align** (por defecto Top), **Anchors**, **CanFocus** (por defecto False), **Cursor**, **Enabled**, **HitTest**, **Locked**, **Margins**, **Opacity**, **PopupMenu**, **Position**, **TabOrder**, **Visible**, **Width** y, a partir de 10.1 Berlin, **TabStop**, **ParentShowHint** y **ShowHint**. Los eventos incluyen OnClick, OnEnter, OnExit, OnKeyDown, OnKeyUp, los eventos del ratón, OnPainting, OnPaint y OnResize.

Métodos

Método	De-vuelve	Descripción
ItemAtPos(X, Y: Single)	Integer	El índice del elemento en un punto del control, CS_TOOLBAR_HIT_OVERFLOW (-2) para el botón de desbordamiento o CS_TOOLBAR_HIT_NONE (-1). Los separadores y los espaciadores nunca reciben el impacto.
ItemRect(aIndex: Integer)	TRectF	El rectángulo de un elemento en coordenadas del control, o un rectángulo vacío cuando el elemento no está en la barra.
OverflowRect	TRectF	El rectángulo del botón de desbordamiento, vacío cuando no hay desbordamiento.
ClickItem(aIndex: Integer)		Activa un elemento como lo haría un clic: invierte un interruptor, dispara OnItemClick, ejecuta el OnClick del elemento o su Action y muestra el menú de un desplegable.
InitiateAction		Permite que cada elemento con una Action se actualice. El framework lo llama en reposo.

Eventos

Evento	Tipo	Descripción
OnItem-Click	TsgcTool-barItemE-vent	Se hizo clic, se tocó o se eligió en el menú de desbordamiento un botón, interruptor o desplegable habilitado, o se activó con Enter o Espacio. Down de un interruptor ya ha cambiado. El OnClick del elemento o su Action se ejecuta después de este evento.

Ejemplo

```
uses
  System.Classes, FMX.Forms, FMX.Types, FMX.Memo,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_FMX;

procedure TMainForm.CreateBars;
var
  oItem: TsgcToolbarItem;
  oPanel: TsgcStatusBarPanel;
begin
  // the toolbar, aligned to the top by default; its height follows ButtonSize
  Toolbar := TsgcFMXToolbar.Create(Self);
  Toolbar.Parent := Self;
  Toolbar.Theme := utDark;
  Toolbar.OnItemClick := ToolbarItemClick;
  oItem := Toolbar.Items.Add;
  oItem.Caption := 'Open';
  oItem.Glyph := 'O';
  oItem := Toolbar.Items.Add;
  oItem.Kind := tikToggle;
  oItem.Caption := 'Wrap';
  oItem.Glyph := 'W';

  // the status bar, aligned to the bottom by default
  StatusBar := TsgcFMXStatusBar.Create(Self);
  StatusBar.Parent := Self;
```

```
  StatusBar.Theme := utDark;
  oPanel := StatusBar.Panels.Add;
  oPanel.Spring := True;
  oPanel.Text := 'Ready';
  oPanel := StatusBar.Panels.Add;
  oPanel.Kind := spkProgress;
  oPanel.Width := 120;

  Memo := TMemo.Create(Self);
  Memo.Parent := Self;
  Memo.Align := TAlignLayout.Client;
end;

procedure TMainForm.ToolbarItemClick(Sender: TObject; aItem: TsgcToolbarItem);
begin
  if aItem.Kind = tikToggle then
    Memo.WordWrap := aItem.Down
  else
    StatusBar.SetPanelText(0, aItem.Caption + ' clicked');
end;
```

Consulte también

- TsgcVCLToolbar
- Inicio rápido FMX
- Plataformas

TsgcVCLStatusBar

Unidad: sgcToolbar_VCL
TsgcVCLStatusBar es la barra de estado VCL, solo para Windows. Se alinea abajo por defecto y muestra sus Panels: paneles de texto, de progreso y de indicador, fijos, AutoSize o Spring. Muestra el Hint del panel que hay bajo el ratón y su tirador de tamaño redimensiona el formulario. Cualquier hilo puede actualizar los paneles con los tres setters seguros entre hilos.

Propiedades

Propiedad	Tipo	Descripción
Panels	TsgcStatusBarPanels	Los paneles, de izquierda a derecha. Consulte TsgcStatusBarPanels.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom, que usa los colores claros. Los colores de los indicadores siguen al tema.
AccentColor	TsgcUIColor	Color de las barras de progreso, como $AARRGGBB. 0 (por defecto) usa el acento del tema.
SizeGrip	Boolean	Dibuja el tirador de tamaño en el extremo derecho. Redimensiona el formulario padre cuando el formulario tiene un borde redimensionable y no está maximizado. Por defecto True. Consulte Tirador de tamaño.
Height	Integer	Altura de la barra, 24 píxeles lógicos por defecto.
Model	TsgcStatusBarModel (public, solo lectura)	El modelo de la barra. Consulte TsgcStatusBarModel.
MaxFPS	Integer	El límite de control de fotogramas de la base de controles de sgcUICore.

Los miembros habituales de un control también están publicados: **Align** (por defecto alBottom), **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **Visible**, OnClick, OnContextPopup, OnMouseDown, OnMouseMove, OnMouseUp y OnResize.

Métodos

Método	Devuelve	Descripción
SetPanelText(aIndex: Integer; const aText: string)		Establece el Text de un panel. Seguro entre hilos, un índice no válido se ignora. Consulte Actualizaciones seguras entre hilos.
SetProgress(aIndex, aValue: Integer)		Establece el Progress (de 0 a 100) de un panel. Seguro entre hilos.
SetIndicator(aIndex: Integer; aState: TsgcStatusIndicatorState)		Establece el IndicatorState de un panel. Seguro entre hilos.
PanelAtPos(X, Y: Integer)	Integer	El índice del panel en un punto del control, o -1.

PanelRect(aIndex: Integer)	TRect	El rectángulo de un panel en coordenadas del control.
GripRect	TRect	El rectángulo del tirador de tamaño, vacío cuando SizeGrip es False.

Eventos

Evento	Tipo	Descripción
OnPanelClick	TsgcStatusBarPanelEvent	Se hizo clic o se tocó un panel. aPanel es el panel.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
procedure TFeatureForm.StatusBarEvents;
begin
  StatusBar.ShowHint := True;          // shows the Hint of the panel under the mouse
  StatusBar.OnPanelClick := StatusPanelClick;
end;

procedure TFeatureForm.StatusPanelClick(Sender: TObject;
  aPanel: TsgcStatusBarPanel);
begin
  if aPanel.Kind = spkIndicator then
    ShowMessage(aPanel.Text);
end;
```

Consulte también

- TsgcFMXStatusBar
- TsgcStatusBarPanel
- Actualizaciones seguras entre hilos
- TsgcStatusBarModel

TsgcFMXStatusBar

Unidad: sgcToolbar_FMX (Delphi XE6 o posterior)
TsgcFMXStatusBar es la barra de estado de FireMonkey con las mismas propiedades, métodos y eventos que TsgcVCLStatusBar. Se alinea abajo por defecto. Los setters seguros entre hilos ponen en cola el repintado en el hilo principal. El tirador de tamaño se dibuja, pero no redimensiona el formulario.

Propiedades

Propiedad	Tipo	Descripción
Panels	TsgcStatusBarPanels	Los paneles, de izquierda a derecha. Consulte TsgcStatusBarPanels.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom, que usa los colores claros. Los colores de los indicadores siguen al tema.
AccentColor	TsgcUIColor	Color de las barras de progreso, como $AARRGGBB. 0 (por defecto) usa el acento del tema.
SizeGrip	Boolean	Dibuja el tirador de tamaño en el extremo derecho. FireMonkey no tiene prueba de impacto en el área no cliente, por lo que el tirador no redimensiona el formulario. Por defecto True. Consulte Tirador de tamaño.
Height	Single	Altura de la barra, 24 píxeles lógicos por defecto.
Model	TsgcStatusBarModel (public, solo lectura)	El modelo de la barra. Consulte TsgcStatusBarModel.
MaxFPS	Integer	El límite de control de fotogramas de la base de controles de sgcUICore.

Los miembros habituales de un control FireMonkey también están publicados: **Align** (por defecto Bottom), **Anchors**, **Cursor**, **Enabled**, **HitTest**, **Locked**, **Margins**, **Opacity**, **PopupMenu**, **Position**, **Visible**, **Width** y, a partir de 10.1 Berlin, **Size**, **ParentShowHint** y **ShowHint**. Los eventos incluyen OnClick, los eventos del ratón, OnPainting, OnPaint y OnResize.

Métodos

Método	Devuelve	Descripción
SetPanelText(aIndex: Integer; const aText: string)		Establece el Text de un panel. Seguro entre hilos, un índice no válido se ignora. Consulte Actualizaciones seguras entre hilos.
SetProgress(aIndex, aValue: Integer)		Establece el Progress (de 0 a 100) de un panel. Seguro entre hilos.
SetIndicator(aIndex: Integer; aState: TsgcStatusIndicatorState)		Establece el IndicatorState de un panel. Seguro entre hilos.
PanelAtPos(X, Y: Single)	Integer	El índice del panel en un punto del control, o -1.

PanelRect(aIndex: Integer)	TRectF	El rectángulo de un panel en coordenadas del control.
GripRect	TRectF	El rectángulo del tirador de tamaño, vacío cuando SizeGrip es False.

Eventos

Evento	Tipo	Descripción
OnPanelClick	TsgcStatusBarPanelEvent	Se hizo clic o se tocó un panel. aPanel es el panel.

Ejemplo

La parte de la barra de estado del inicio rápido de FireMonkey: un panel de texto elástico y un panel de progreso.

```
uses
  System.Classes, FMX.Forms, FMX.Types, FMX.Memo,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_FMX;

procedure TMainForm.CreateBars;
var
  oItem: TsgcToolbarItem;
  oPanel: TsgcStatusBarPanel;
begin
  // the toolbar, aligned to the top by default; its height follows ButtonSize
  Toolbar := TsgcFMXToolbar.Create(Self);
  Toolbar.Parent := Self;
  Toolbar.Theme := utDark;
  Toolbar.OnItemClick := ToolbarItemClick;
  oItem := Toolbar.Items.Add;
  oItem.Caption := 'Open';
  oItem.Glyph := 'O';
  oItem := Toolbar.Items.Add;
  oItem.Kind := tikToggle;
  oItem.Caption := 'Wrap';
  oItem.Glyph := 'W';

  // the status bar, aligned to the bottom by default
  StatusBar := TsgcFMXStatusBar.Create(Self);
  StatusBar.Parent := Self;
  StatusBar.Theme := utDark;
  oPanel := StatusBar.Panels.Add;
  oPanel.Spring := True;
  oPanel.Text := 'Ready';
  oPanel := StatusBar.Panels.Add;
  oPanel.Kind := spkProgress;
  oPanel.Width := 120;

  Memo := TMemo.Create(Self);
  Memo.Parent := Self;
  Memo.Align := TAlignLayout.Client;
end;

procedure TMainForm.ToolbarItemClick(Sender: TObject; aItem: TsgcToolbarItem);
begin
  if aItem.Kind = tikToggle then
    Memo.WordWrap := aItem.Down
  else
    StatusBar.SetPanelText(0, aItem.Caption + ' clicked');
end;
```

Consulte también

- TsgcVCLStatusBar
- Inicio rápido FMX
- Tirador de tamaño

TsgcToolbarItem

Unidad: sgcToolbar_Model
TsgcToolbarItem es un elemento de una barra de herramientas, un TCollectionItem de TsgcToolbarItems. La misma clase la usan las barras de herramientas VCL y FMX.

Propiedades

Propiedad	Tipo	Descripción
Kind	TsgcToolbarItemKind	tikButton (por defecto), tikToggle, tikDropdown, tikSeparator, tikSpacer o tikLabel. Consulte Tipos de elemento.
Caption	string	El texto del elemento.
Glyph	string	Un texto corto dibujado en el hueco del icono cuando el elemento no tiene imagen, por ejemplo B para Bold.
ImageIndex	Integer	Índice en Images de la barra, -1 (por defecto) para ninguna.
Hint	string	El hint del elemento. Cuando está vacío y los títulos están ocultos, el hint es el Caption.
Enabled	Boolean	Un elemento deshabilitado se dibuja atenuado e ignora el ratón y el teclado. Por defecto True.
Visible	Boolean	Un elemento oculto no ocupa espacio. Por defecto True.
Down	Boolean	El estado de un elemento tikToggle. Con GroupIndex mayor que 0, establecer Down desactiva los demás elementos del mismo grupo. Por defecto False.
GroupIndex	Integer	Los interruptores con el mismo GroupIndex mayor que 0 forman un grupo de opciones. Por defecto 0. Consulte Interruptores y grupos de opciones.
Width	Integer	Ancho en píxeles lógicos, 0 (por defecto) para automático. En un espaciador es el ancho mínimo.
Tag	Integer	Un valor libre para su código.
DropdownMenu	TComponent	El menú emergente de un elemento tikDropdown: un TPopupMenu de VCL en TsgcVCLToolbar, un TPopupMenu de FireMonkey en TsgcFMXToolbar. La barra borra la referencia cuando se libera el menú. Consulte Menús desplegables.
Action	TBasicAction	Una acción vinculada. Caption, Enabled, Visible, Hint, ImageIndex y Checked (Down de un interruptor) la siguen, y un clic la ejecuta salvo que OnClick esté asignado. No está en la edición Community. Consulte Vínculos de acción.
BoundsRect	TsgcUIRect (public, solo lectura)	El rectángulo del último Layout, en píxeles de dispositivo.
Shown	Boolean (public, solo lectura)	True cuando el último Layout colocó el elemento en la barra.

InOverflow	Boolean (public, solo lectura)	True cuando el último Layout movió el elemento a la lista de desbordamiento.

Métodos

Método	Devuelve	Descripción
Click(aComponent: TComponent = nil)		Ejecuta OnClick cuando está asignado y, si no, la Action vinculada con aComponent como su ActionComponent. No invierte un interruptor ni dispara OnItemClick, use ClickItem de la barra para eso.
InitiateAction		Permite que la Action vinculada se actualice. No hace nada sin una Action.
IsButton	Boolean	True para los tipos en los que se puede hacer clic: tikButton, tikToggle y tikDropdown.

Eventos

Evento	Tipo	Descripción
OnClick	TNotifyEvent	Se activó el elemento. Sender es el elemento. Se ejecuta después de OnItemClick de la barra.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
var
  oItem: TsgcToolbarItem;
begin
  oItem := Toolbar.Items.Add;          // tikButton is the default kind
  oItem.Caption := 'Save';
  oItem.Glyph := 'S';

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikToggle;             // stays down until clicked again
  oItem.Caption := 'Bold';
  oItem.Glyph := 'B';

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikDropdown;           // shows its DropdownMenu
  oItem.Caption := 'Zoom';

  Toolbar.Items.Add.Kind := tikSeparator;  // a vertical line

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikLabel;              // text only, never clicked
  oItem.Caption := 'Find:';

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikSpacer;             // shares the free width
  oItem.Width := 16;                   // minimum width, logical pixels
end;
```

Consulte también

- TsgcToolbarItems
- Tipos de elemento
- Interruptores y grupos de opciones

TsgcToolbarItems

Unidad: sgcToolbar_Model
TsgcToolbarItems es la colección de TsgcToolbarItem de una barra de herramientas y de TsgcToolbarModel. Un cambio en un elemento actualiza la barra.

Propiedades

Propiedad	Tipo	Descripción
Items[Index: Integer]	TsgcToolbarItem	El elemento en Index. Es la propiedad por defecto, por lo que Toolbar.Items[0] es el primer elemento.
Count	Integer	El número de elementos.

Métodos

Método	Devuelve	Descripción
Add	TsgcToolbarItem	Añade un elemento tikButton al final y lo devuelve.
Delete(Index: Integer)		Elimina un elemento.
Clear		Elimina todos los elementos.

Para cambiar muchos elementos con un solo repintado, encierre los cambios entre BeginUpdate y EndUpdate del modelo.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
var
  i: Integer;
begin
  // disables every clickable item with Tag 1, in one repaint
  Toolbar.Model.BeginUpdate;
  try
    for i := 0 to Toolbar.Items.Count - 1 do
      if Toolbar.Items[i].IsButton and (Toolbar.Items[i].Tag = 1) then
        Toolbar.Items[i].Enabled := False;
  finally
    Toolbar.Model.EndUpdate;
  end;
  Toolbar.Items.Delete(Toolbar.Items.Count - 1);
end;
```

Consulte también

- TsgcToolbarItem
- TsgcToolbarModel

TsgcStatusBarPanel

Unidad: sgcToolbar_Model
TsgcStatusBarPanel es un panel de una barra de estado, un TCollectionItem de TsgcStatusBarPanels. Text, Progress e IndicatorState están protegidos por el bloqueo del modelo.

Propiedades

Propiedad	Tipo	Descripción
Kind	TsgcStatusPanelKind	spkText (por defecto), spkProgress o spkIndicator. Consulte Progreso e indicadores.
Text	string	El texto de un panel de texto o indicador.
Width	Integer	Ancho en píxeles lógicos cuando no se establece AutoSize ni Spring. Por defecto 100.
AutoSize	Boolean	El ancho sigue al texto y al punto de un indicador. Por defecto False.
Spring	Boolean	Los paneles elásticos se reparten el ancho que dejan los demás. Por defecto False. Consulte Paneles de estado.
Alignment	TAlignment	taLeftJustify (por defecto), taCenter o taRightJustify.
Progress	Integer	El valor de un panel spkProgress, de 0 a 100. Por defecto 0.
IndicatorState	TsgcStatusIndicatorState	El estado de un panel spkIndicator: sisOff (por defecto), sisOk, sisWarning o sisError.
Hint	string	El hint que se muestra sobre el panel.
Tag	Integer	Un valor libre para su código.
BoundsRect	TsgcUIRect (public, solo lectura)	El rectángulo del último Layout, en píxeles de dispositivo.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
var
  oPanel: TsgcStatusBarPanel;
begin
  StatusBar.Panels.Clear;

  oPanel := StatusBar.Panels.Add;      // Spring: takes the width that is left
  oPanel.Spring := True;
  oPanel.Text := 'Ready';

  oPanel := StatusBar.Panels.Add;      // AutoSize: as wide as its text
  oPanel.AutoSize := True;
  oPanel.Text := 'UTF-8';

  oPanel := StatusBar.Panels.Add;      // fixed: Width logical pixels
  oPanel.Width := 110;
```

```
  oPanel.Alignment := taRightJustify;
  oPanel.Text := 'Ln 1, Col 1';
  oPanel.Hint := 'Caret position';
end;
```

Consulte también

- TsgcStatusBarPanels
- Paneles de estado
- Progreso e indicadores

TsgcStatusBarPanels

Unidad: sgcToolbar_Model
TsgcStatusBarPanels es la colección de TsgcStatusBarPanel de una barra de estado y de TsgcStatusBarModel.

Propiedades

Propiedad	Tipo	Descripción
Items[Index: Integer]	TsgcStatusBarPanel	El panel en Index. Es la propiedad por defecto, por lo que StatusBar.Panels[0] es el primer panel.
Count	Integer	El número de paneles.

Métodos

Método	Devuelve	Descripción
Add	TsgcStatusBarPanel	Añade un panel spkText al final y lo devuelve.
Delete(Index: Integer)		Elimina un panel.
Clear		Elimina todos los paneles.

Añada, elimine y configure los paneles desde el hilo principal. Desde un hilo de trabajo use SetPanelText, SetProgress y SetIndicator de la barra, consulte Actualizaciones seguras entre hilos.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
var
  i: Integer;
begin
  // resets every indicator panel
  for i := 0 to StatusBar.Panels.Count - 1 do
    if StatusBar.Panels[i].Kind = spkIndicator then
      StatusBar.SetIndicator(i, sisOff);
end;
```

Consulte también

- TsgcStatusBarPanel
- TsgcStatusBarModel

TsgcToolbarModel

Unidad: sgcToolbar_Model
TsgcToolbarModel es el modelo independiente del framework de una barra de herramientas. Cada barra de herramientas posee uno, disponible mediante su propiedad **Model**, y un modelo creado en código dibuja una barra de herramientas sin ventana. Los tamaños son píxeles lógicos. Layout los multiplica por la escala, y cada rectángulo calculado está en píxeles de dispositivo.

Propiedades

Propiedad	Tipo	Descripción
Items, ShowCaptions, ButtonSize, Spacing, Padding, Theme, AccentColor		Las propiedades publicadas de la barra de herramientas, consulte TsgcVCLToolbar.
HotIndex, PressedIndex, FocusedIndex	Integer	El estado de interacción: un índice de elemento, CS_TOOLBAR_HIT_OVERFLOW para el botón de desbordamiento, o -1.
HasOverflow	Boolean (solo lectura)	True cuando el último Layout movió elementos a la lista de desbordamiento.
OverflowCount	Integer (solo lectura)	El número de elementos de la lista de desbordamiento.
OverflowItem[Index: Integer]	TsgcToolbarItem (solo lectura)	Un elemento de la lista de desbordamiento.
OverflowRect	TsgcUIRect (solo lectura)	El botón de desbordamiento del último Layout, en píxeles de dispositivo.
Owner	TPersistent	El componente propietario del modelo, usado para la ruta de nombres en tiempo de diseño.

Métodos

Método	Devuelve	Descripción
Layout(aWidth, aHeight: Integer; aScale: Double; aMeasure or aCanvas)		Calcula los rectángulos de los elementos para una barra de aWidth por aHeight píxeles de dispositivo. aMeasure (un TsgcToolbarMeasureEvent) o aCanvas mide el texto; un measure nil usa una estimación de 0,55 veces el tamaño de la fuente por carácter. Los espaciadores se reparten el ancho libre. Con el menú de desbordamiento, los elementos que no caben pasan a la lista de desbordamiento; si no, se recortan.

HitTest(aX, aY: Double)	Integer	El elemento en un punto del último Layout, CS_TOOLBAR_HIT_OVERFLOW o CS_TOOLBAR_HIT_NONE.
Render(aCanvas; aWidth, aHeight: Integer; aScale: Double = 1)		Pinta la barra en un lienzo de sgcUICore, usando OnGetImage para las imágenes.
SaveToPNG, SaveToSVG(aFileName or aStream; aWidth, aHeight: Integer; aScale: Double = 1)		Escribe la barra como imagen. Consulte Renderizado sin interfaz.
BeginUpdate, EndUpdate		Agrupa los cambios en un único OnChange.
GetAccent	TsgcUIColor	AccentColor, o el acento del tema cuando es 0.
ShowsCaption(aItem)	Boolean	True cuando el elemento dibuja su título: ShowCaptions, o el elemento no tiene icono.
HasIcon(aItem)	Boolean	True cuando el elemento tiene un Glyph o un ImageIndex.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El modelo cambió. Las barras se repintan con él.
OnGetImage	TsgcToolbarImageEvent	Devuelve la imagen de un ImageIndex como TsgcUIImage para Render y la salida sin interfaz.

Ejemplo

Layout y HitTest sin un control, como lo hace una prueba o un servidor.

```
uses
  SysUtils,
  sgcUICore_Theme, sgcToolbar_Model;

var
  oModel: TsgcToolbarModel;
  vMeasure: TsgcToolbarMeasureEvent;
  vIndex: Integer;
begin
  oModel := TsgcToolbarModel.Create;
  try
    oModel.BeginUpdate;
    try
      oModel.ShowCaptions := False;
      oModel.ButtonSize := 28;
      oModel.Items.Add.Caption := 'Cut';
      oModel.Items.Add.Caption := 'Copy';
      oModel.Items.Add.Caption := 'Paste';
      oModel.Items[0].Glyph := 'X';
      oModel.Items[1].Glyph := 'C';
      oModel.Items[2].Glyph := 'V';
    finally
      oModel.EndUpdate;
```

```
    end;
    // nil measures the text with an estimate; a TsgcUICanvas measures it exactly
    vMeasure := nil;
    oModel.Layout(200, 40, 1, vMeasure);
    vIndex := oModel.HitTest(50, 20);
    if vIndex >= 0 then
      WriteLn('hit ', oModel.Items[vIndex].Caption, ' from x=',
        oModel.Items[vIndex].BoundsRect.Left:0:0);
    WriteLn('overflow items: ', oModel.OverflowCount);
  finally
    oModel.Free;
  end;
end;
```

Consulte también

- TsgcVCLToolbar
- Renderizado sin interfaz
- Arquitectura

TsgcStatusBarModel

Unidad: sgcToolbar_Model
TsgcStatusBarModel es el modelo independiente del framework de una barra de estado. Cada barra de estado posee uno, disponible mediante su propiedad **Model**. Sus tres setters son seguros entre hilos: toman el bloqueo del modelo, cambian el valor y disparan OnChange desde el hilo que los llama, y solo cuando el valor ha cambiado. Cualquier otra propiedad está pensada para el hilo principal.

Propiedades

Propiedad	Tipo	Descripción
Panels, Theme, AccentColor, SizeGrip		Las propiedades publicadas de la barra de estado, consulte TsgcVCLStatusBar.
GripRect	TsgcUIRect (solo lectura)	El tirador de tamaño del último Layout en píxeles de dispositivo, vacío cuando SizeGrip es False.
Owner	TPersistent	El componente propietario del modelo, usado para la ruta de nombres en tiempo de diseño.

Métodos

Método	Devuelve	Descripción
SetPanelText, SetProgress, SetIndicator		Los setters seguros entre hilos. Un índice no válido se ignora.
Layout(aWidth, aHeight: Integer; aScale: Double; aMeasure or aCanvas)		Calcula los rectángulos de los paneles: primero los anchos fijos, después los paneles AutoSize según su texto, y luego los paneles Spring se reparten lo que queda, nunca por debajo de 0. El tirador de tamaño se reserva a la derecha cuando SizeGrip es True.
HitTest(aX, aY: Double)	Integer	El panel en un punto del último Layout, o -1.
Render, SaveToPNG, SaveToSVG		Pintan la barra en un lienzo o la escriben como imagen, igual que en TsgcToolbarModel.
GetIndicatorColor(aState)	TsgcUIColor	El color de un estado de indicador en el tema actual.
GetAccent	TsgcUIColor	AccentColor, o el acento del tema cuando es 0.
BeginUpdate, EndUpdate		Agrupa los cambios en un único OnChange.
Lock, Unlock		El bloqueo que protege Text, Progress e IndicatorState.

Eventos

Evento	Tipo	Descripción
OnChan-ge	TNotif-yEvent	El modelo cambió. Tras un setter seguro entre hilos se dispara en el hilo que llama, y las barras trasladan el repintado al hilo principal.

Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcToolbar_Model;

var
  oModel: TsgcStatusBarModel;
begin
  oModel := TsgcStatusBarModel.Create;
  try
    oModel.Theme := utDark;
    oModel.SizeGrip := False;
    oModel.Panels.Add.Spring := True;
    oModel.Panels.Add.Kind := spkIndicator;
    // thread-safe setters: an index out of range is ignored
    oModel.SetPanelText(0, 'Connected to feed');
    oModel.SetIndicator(1, sisWarning);
    WriteLn('warning colour: ', IntToHex(oModel.GetIndicatorColor(sisWarning), 8));
    oModel.SaveToSVG('status_dark.svg', 400, 24);
  finally
    oModel.Free;
  end;
end;
```

Consulte también

- TsgcVCLStatusBar
- Actualizaciones seguras entre hilos
- Renderizado sin interfaz

Tipos de elemento

El **Kind** de un TsgcToolbarItem decide cómo se dibuja y cómo reacciona. Los elementos se colocan de izquierda a derecha en el orden de la colección, con Spacing píxeles lógicos entre ellos.

Tipo	Comportamiento
tikBut-ton	El valor por defecto. Un clic dispara OnItemClick de la barra y después OnClick del elemento o su Action.
tikTog-gle	Un botón que se queda pulsado. Un clic invierte Down antes de que se dispare OnItemClick. Con GroupIndex mayor que 0 es un botón de opción, consulte Interruptores y grupos de opciones.
tikDrop-down	Un botón con una pequeña flecha que muestra su DropdownMenu bajo el botón. Consulte Menús desplegables.
tikSepa-rator	Una línea vertical fina. Nunca recibe clics.
tikSpa-cer	Un elemento vacío que ocupa el ancho libre de la barra, compartido con los demás espaciadores. Empuja a la derecha los elementos que le siguen. Width es su ancho mínimo.
tikLabel	Un título sin botón, por ejemplo antes de un campo de edición. Nunca recibe clics.

Un botón dibuja su icono (la imagen de ImageIndex, o el texto de Glyph) y su Caption. Con ShowCaptions en False solo se dibuja el icono y el título pasa a ser el hint. Un elemento sin icono siempre dibuja su título. Un Width mayor que 0 fija el ancho de un elemento en píxeles lógicos.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

var
  oItem: TsgcToolbarItem;
begin
  oItem := Toolbar.Items.Add;          // tikButton is the default kind
  oItem.Caption := 'Save';
  oItem.Glyph := 'S';

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikToggle;             // stays down until clicked again
  oItem.Caption := 'Bold';
  oItem.Glyph := 'B';

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikDropdown;           // shows its DropdownMenu
  oItem.Caption := 'Zoom';

  Toolbar.Items.Add.Kind := tikSeparator;  // a vertical line

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikLabel;              // text only, never clicked
  oItem.Caption := 'Find:';
```

```
  oItem := Toolbar.Items.Add;
  oItem.Kind := tikSpacer;              // shares the free width
  oItem.Width := 16;                    // minimum width, logical pixels
end;
```

Consulte también

- TsgcToolbarItem
- Imágenes y glifos
- Desbordamiento

Interruptores y grupos de opciones

Un elemento **tikToggle** tiene un estado **Down**, dibujado con un tinte de acento. Un clic invierte Down y después dispara OnItemClick, por lo que el manejador lee el nuevo estado.
Los interruptores que comparten un **GroupIndex** mayor que 0 forman un grupo de opciones: establecer Down en uno de ellos, con un clic o desde código, desactiva los demás. Un clic en un interruptor de opción que ya está pulsado lo deja pulsado, de modo que un elemento del grupo permanece seleccionado. Un interruptor vinculado a una Action sigue el Checked de la acción.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

procedure TFeatureForm.Toggles;
const
  CAPTIONS: array [0 .. 2] of string = ('Left', 'Center', 'Right');
var
  i: Integer;
  oItem: TsgcToolbarItem;
begin
  // three toggles with the same GroupIndex behave as radio buttons
  for i := 0 to 2 do
  begin
    oItem := Toolbar.Items.Add;
    oItem.Kind := tikToggle;
    oItem.GroupIndex := 1;
    oItem.Caption := CAPTIONS[i];
    oItem.Tag := i;
  end;
  Toolbar.Items[0].Down := True;       // clears the other items of group 1
  Toolbar.OnItemClick := ToolbarItemClick;
end;

procedure TFeatureForm.ToolbarItemClick(Sender: TObject;
  aItem: TsgcToolbarItem);
const
  ALIGNMENTS: array [0 .. 2] of TAlignment = (taLeftJustify, taCenter,
    taRightJustify);
begin
  // Down has already been flipped when OnItemClick fires
  if (aItem.Kind = tikToggle) and (aItem.GroupIndex = 1) and aItem.Down then
    Memo.Alignment := ALIGNMENTS[aItem.Tag];
end;
```

Consulte también

- Tipos de elemento
- Vínculos de acción
- TsgcToolbarItem

Menús desplegables

Un elemento **tikDropdown** dibuja una pequeña flecha tras su título y muestra el menú emergente de su propiedad **DropdownMenu** bajo el botón: un TPopupMenu de VCL en TsgcVCLToolbar, un TPopupMenu de FireMonkey en TsgcFMXToolbar. La propiedad se declara como TComponent para que el elemento siga siendo independiente del framework. La barra borra la referencia cuando se libera el menú.
En VCL el menú se abre al pulsar el botón. OnItemClick y el OnClick del elemento se disparan primero, por lo que un manejador puede rellenar o marcar el menú antes de que se muestre. Un elemento desplegable de la lista de desbordamiento muestra su menú bajo el botón de desbordamiento.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

procedure TFeatureForm.Dropdown;
var
  i: Integer;
  oMenuItem: TMenuItem;
  oItem: TsgcToolbarItem;
begin
  pmZoom := TPopupMenu.Create(Self);
  for i := 1 to 4 do
  begin
    oMenuItem := TMenuItem.Create(pmZoom);
    oMenuItem.Caption := IntToStr(i * 50) + '%';
    oMenuItem.Tag := i * 50;
    oMenuItem.OnClick := ZoomClick;
    pmZoom.Items.Add(oMenuItem);
  end;

  oItem := Toolbar.Items.Add;
  oItem.Kind := tikDropdown;
  oItem.Caption := 'Zoom';
  oItem.Glyph := 'Z';
  oItem.DropdownMenu := pmZoom;        // a VCL TPopupMenu
end;

procedure TFeatureForm.ZoomClick(Sender: TObject);
begin
  Memo.Font.Size := 10 * TMenuItem(Sender).Tag div 100;
end;
```

Consulte también

- Tipos de elemento
- Desbordamiento
- Solución de problemas

Desbordamiento

Cuando los elementos no caben en el ancho de la barra, el Layout mueve los últimos a la lista de desbordamiento y reserva un botón de desbordamiento, dibujado con una doble punta de flecha, en el extremo derecho. Al pulsarlo se muestra el **OverflowMenu** de la barra, un menú emergente construido en tiempo de ejecución con una entrada por cada elemento de la lista. Elegir una entrada activa ese elemento exactamente igual que un clic: un interruptor se invierte, se dispara OnItemClick y un desplegable muestra su menú.
La lista de desbordamiento del último Layout está disponible mediante el modelo: **HasOverflow**, **OverflowCount**, **OverflowItem**[i] y la propiedad **InOverflow** de cada elemento. ItemAtPos devuelve CS_TOOLBAR_HIT_OVERFLOW sobre el botón de desbordamiento.

Edición Community

El menú de desbordamiento no forma parte de la edición Community (SGC_TOOLBAR_ADVANCED). En ella los elementos que no caben se recortan en el borde derecho y la propiedad OverflowMenu no existe. Consulte Ediciones.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

var
  i: Integer;
begin
  Toolbar.Width := 240;
  for i := 1 to 12 do
    Toolbar.Items.Add.Caption := 'Command ' + IntToStr(i);
  // after the next layout the items that do not fit are in the overflow list
  if Toolbar.Model.HasOverflow then
    StatusBar.SetPanelText(0, IntToStr(Toolbar.Model.OverflowCount) +
      ' items in the overflow menu');
end;
```

Consulte también

- TsgcToolbarModel
- Menús desplegables
- Ediciones

Vínculos de acción

La propiedad **Action** vincula un elemento a cualquier TBasicAction, un TAction de VCL o de FireMonkey, sin que el modelo dependa de ninguno de los dos frameworks. El elemento sigue entonces el **Caption**, **Enabled**, **Visible**, **Hint**, **ImageIndex** y **Checked** de la acción, donde Checked controla Down de un elemento tikToggle. Un TAction de FireMonkey da su Text como Caption.
Un clic ejecuta la acción, con la barra como su ActionComponent, salvo que el elemento tenga un manejador OnClick. Las barras llaman a InitiateAction en reposo, por lo que se ejecuta el OnUpdate de cada acción vinculada y una acción deshabilitada deshabilita su botón. A partir de Delphi XE3 la acción envía sus cambios al elemento. Antes de XE3 el elemento copia las propiedades de la acción en reposo.

Edición Community

Los vínculos de acción no forman parte de la edición Community (SGC_TOOLBAR_ADVANCED): la propiedad Action no existe. Use en su lugar OnClick del elemento u OnItemClick de la barra. Consulte Ediciones.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

procedure TFeatureForm.Actions;
var
  oItem: TsgcToolbarItem;
begin
  ActionList := TActionList.Create(Self);
  actSave := TAction.Create(Self);
  actSave.ActionList := ActionList;
  actSave.Caption := 'Save';
  actSave.Hint := 'Save the document';
  actSave.OnExecute := SaveExecute;

  oItem := Toolbar.Items.Add;
  oItem.Glyph := 'S';
  // Caption, Hint, Enabled, Visible, ImageIndex and Checked follow the action
  oItem.Action := actSave;
  actSave.Enabled := False;            // the button is disabled too
end;

procedure TFeatureForm.SaveExecute(Sender: TObject);
begin
  Memo.Lines.SaveToFile('document.txt');
end;
```

Consulte también

- TsgcToolbarItem
- Interruptores y grupos de opciones
- Solución de problemas

Imágenes y glifos

Cada elemento en el que se puede hacer clic tiene un hueco de icono de 16 píxeles lógicos. Muestra la imagen en **ImageIndex** de la lista **Images** de la barra o, cuando el elemento no tiene imagen, su **Glyph**: un texto corto como una letra o un símbolo. Un elemento sin ninguno de los dos solo muestra su título.

- En VCL, Images es un TCustomImageList. Cada imagen se convierte una sola vez a ARGB directo, con el alfa recuperado, y se guarda en caché hasta que la lista de imágenes cambia.
- En FMX, Images es el TCustomImageList de FireMonkey de FMX.ImgList y necesita Delphi XE8 o posterior. Las imágenes se toman al tamaño de dispositivo del fotograma. En versiones anteriores los elementos dibujan su Glyph.
- La salida sin interfaz dibuja el Glyph salvo que el evento OnGetImage del modelo devuelva una imagen.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

var
  oBitmap: TBitmap;
  oItem: TsgcToolbarItem;
begin
  ImageList := TImageList.Create(Self);
  ImageList.Width := 16;
  ImageList.Height := 16;
  oBitmap := TBitmap.Create;
  try
    oBitmap.SetSize(16, 16);
    oBitmap.Canvas.Brush.Color := clNavy;
    oBitmap.Canvas.FillRect(Rect(0, 0, 16, 16));
    ImageList.Add(oBitmap, nil);
  finally
    oBitmap.Free;
  end;

  Toolbar.Images := ImageList;
  oItem := Toolbar.Items.Add;
  oItem.Caption := 'Print';
  oItem.Glyph := 'P';                  // drawn only when there is no image
  oItem.ImageIndex := 0;
end;
```

Consulte también

- Tipos de elemento
- Renderizado sin interfaz
- Plataformas

Paneles de estado

Una barra de estado distribuye sus Panels de izquierda a derecha en tres pasadas:

- Los paneles **fijos**, sin AutoSize ni Spring, toman su **Width** en píxeles lógicos (100 por defecto).
- Los paneles **AutoSize** toman el ancho de su texto, más el punto de un indicador.
- Los paneles **Spring** se reparten el ancho que queda, nunca por debajo de 0. Use un panel elástico para el mensaje principal.

Una línea fina separa los paneles. El texto de un panel se alinea con **Alignment**, y **Hint** se muestra cuando el ratón se detiene sobre el panel y ShowHint es True. OnPanelClick informa de un clic en un panel. Cuando SizeGrip es True, el tirador se reserva en el extremo derecho, consulte Tirador de tamaño.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

var
  oPanel: TsgcStatusBarPanel;
begin
  StatusBar.Panels.Clear;

  oPanel := StatusBar.Panels.Add;      // Spring: takes the width that is left
  oPanel.Spring := True;
  oPanel.Text := 'Ready';

  oPanel := StatusBar.Panels.Add;      // AutoSize: as wide as its text
  oPanel.AutoSize := True;
  oPanel.Text := 'UTF-8';

  oPanel := StatusBar.Panels.Add;      // fixed: Width logical pixels
  oPanel.Width := 110;
  oPanel.Alignment := taRightJustify;
  oPanel.Text := 'Ln 1, Col 1';
  oPanel.Hint := 'Caret position';
end;
```

Consulte también

- TsgcStatusBarPanel
- Progreso e indicadores
- Actualizaciones seguras entre hilos

Progreso e indicadores

Además de los paneles de texto (spkText), un panel puede ser una barra de progreso o un indicador de estado.

Tipo	Dibuja
spkProgress	Una barra rellena hasta el porcentaje de **Progress** (de 0 a 100), en el AccentColor de la barra de estado o el acento del tema.
spkIndicator	Un punto de color seguido del **Text** del panel. El color procede de **IndicatorState**, que resulta adecuado para el estado de una conexión o de un flujo de datos.

IndicatorState	Tema claro	Tema oscuro
sisOff	El color de texto atenuado del tema	El color de texto atenuado del tema
sisOk	Verde $FF1A7F37	Verde $FF3FB950
sisWarning	Ámbar $FFBF8700	Ámbar $FFD29922
sisError	Rojo $FFCF222E	Rojo $FFF85149

GetIndicatorColor de TsgcStatusBarModel devuelve el color de un estado en el tema actual.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

var
  oPanel: TsgcStatusBarPanel;
begin
  oPanel := StatusBar.Panels.Add;
  oPanel.Kind := spkProgress;          // a bar in the accent colour
  oPanel.Width := 140;
  oPanel.Progress := 25;               // 0..100

  oPanel := StatusBar.Panels.Add;
  oPanel.Kind := spkIndicator;         // a coloured dot before the text
  oPanel.AutoSize := True;
  oPanel.Text := 'Feed';
  oPanel.IndicatorState := sisWarning; // sisOff, sisOk, sisWarning, sisError

  StatusBar.AccentColor := $FF8250DF;  // progress colour, $AARRGGBB
end;
```

Consulte también

- Paneles de estado
- Actualizaciones seguras entre hilos
- Temas

Actualizaciones seguras entre hilos

Una aplicación en vivo actualiza su barra de estado desde hilos de trabajo: una descarga, un flujo de datos, un cálculo largo. **SetPanelText**, **SetProgress** y **SetIndicator** de la barra de estado pueden llamarse desde cualquier hilo, sin Synchronize ni Queue.

- El modelo protege el Text, Progress e IndicatorState de los paneles con un bloqueo.
- El setter dispara OnChange del modelo desde el hilo que llama, y solo cuando el valor ha cambiado, por lo que repetir el mismo valor no cuesta nada.
- La barra responde con una solicitud de repintado. Desde un hilo de trabajo la barra VCL solo envía un mensaje a su ventana y la barra FMX pone en cola el repintado en el hilo principal, por lo que el framework nunca se toca fuera del hilo principal.
- Un índice fuera de rango se ignora, por lo que un hilo que sobrevive a un panel no genera una excepción.

Cualquier otra propiedad, y añadir, eliminar o configurar paneles, debe hacerse desde el hilo principal. Asegúrese de que el hilo de trabajo se detiene antes de destruir el formulario que posee la barra de estado.

Ejemplo

Un hilo de trabajo que simula una descarga de 50 fragmentos, como en la demo LiveStatus. El panel 0 es un panel de texto elástico, el panel 1 un panel de progreso y el panel 2 un indicador.

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

type
  TDownloadWorker = class(TThread)
  private
    FStatusBar: TsgcVCLStatusBar;
  protected
    procedure Execute; override;
  public
    constructor Create(aStatusBar: TsgcVCLStatusBar);
  end;

constructor TDownloadWorker.Create(aStatusBar: TsgcVCLStatusBar);
begin
  FStatusBar := aStatusBar;
  FreeOnTerminate := True;
  inherited Create(False);
end;

procedure TDownloadWorker.Execute;
var
  i: Integer;
begin
  // panel 0 text, panel 1 progress, panel 2 indicator
  FStatusBar.SetIndicator(2, sisOk);
  for i := 1 to 50 do
  begin
    if Terminated then
      Exit;
    Sleep(100);
    // no Synchronize: the setters are thread-safe and the bar
    // repaints in the main thread
    FStatusBar.SetPanelText(0, Format('Chunk %d of 50', [i]));
    FStatusBar.SetProgress(1, i * 2);
  end;
  FStatusBar.SetIndicator(2, sisOff);
end;

procedure TFeatureForm.StartWorker;
```

```
begin
  TDownloadWorker.Create(StatusBar);
end;
```

Consulte también

- TsgcStatusBarModel
- Progreso e indicadores
- Demos

Tirador de tamaño

Con **SizeGrip** en True, el valor por defecto, la barra de estado reserva una pequeña zona en su extremo derecho y dibuja allí los puntos del tirador. **GripRect** devuelve su rectángulo, vacío cuando SizeGrip es False.

- En VCL, TsgcVCLStatusBar responde a la prueba de impacto en el área no cliente del tirador, por lo que arrastrarlo redimensiona el formulario padre. Lo hace solo cuando el formulario tiene un borde redimensionable y no está maximizado, y nunca en tiempo de diseño.
- En FMX, TsgcFMXStatusBar solo dibuja el tirador. FireMonkey no tiene prueba de impacto en el área no cliente, por lo que el tirador no redimensiona el formulario. Establezca SizeGrip en False cuando el formulario no sea redimensionable.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

var
  vGrip: TRect;
begin
  // the grip resizes the form when the form has a sizeable border
  BorderStyle := bsSizeable;
  StatusBar.SizeGrip := True;
  vGrip := StatusBar.GripRect;         // client rectangle, empty without grip
  Caption := Format('Grip at %d,%d', [vGrip.Left, vGrip.Top]);
end;
```

Consulte también

- Paneles de estado
- TsgcVCLStatusBar

Teclado

Se llega a la barra de herramientas con la tecla Tab cuando **TabStop** es True en VCL, o **CanFocus** es True en FMX. Ambos son False por defecto, por lo que la barra de herramientas no le quita el foco al editor que hay debajo. Cuando la barra de herramientas tiene el foco, el botón con foco muestra un anillo de foco:

Tecla	Acción
Left, Right	Pasa al botón, interruptor o desplegable habilitado anterior o siguiente.
Home, End	Pasa al primero o al último.
Enter, Espacio	Activa el elemento con foco, como lo haría un clic.

ClickItem activa un elemento desde código, por ejemplo desde un atajo de teclado del formulario. La barra de estado no tiene interfaz de teclado.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

begin
  // Tab moves the focus to the bar; Left, Right, Home and End move between
  // the buttons, Enter and Space activate the focused one
  Toolbar.TabStop := True;
  Toolbar.TabOrder := 0;
  // activates the first item from code, exactly as a click would
  Toolbar.ClickItem(0);
end;
```

Consulte también

- TsgcVCLToolbar
- TsgcFMXToolbar

Temas

Las barras toman sus colores de los tokens de tema de sgcUICore, los mismos que usan los demás productos de la suite. **Theme** es utLight (por defecto), utDark o utCustom, que usa los colores claros. Cada barra tiene su propio Theme, por lo que debe establecerse en la barra de herramientas y en la barra de estado.

- El fondo es el color Surface, con una línea Border bajo la barra de herramientas y sobre la barra de estado.
- Los botones resaltados y pulsados reciben una superposición, un interruptor pulsado un tinte de acento, el botón con foco un anillo de foco y un elemento deshabilitado el color de texto atenuado.
- **AccentColor**, como $AARRGGBB, sustituye el acento del tema: el anillo de foco y el tinte de pulsado en la barra de herramientas, las barras de progreso en la barra de estado. 0 usa el acento del tema.
- Los colores de los indicadores siguen al tema, consulte Progreso e indicadores.

Ejemplo

El ejemplo es un método de un formulario VCL con un **Toolbar** (TsgcVCLToolbar), un **StatusBar** (TsgcVCLStatusBar) y un **Memo** (TMemo).

```
uses
  Windows, Types, Forms, Classes, SysUtils, Controls, StdCtrls, Menus,
  ActnList, ImgList, Graphics, Dialogs,
  sgcUICore_Theme, sgcToolbar_Model, sgcToolbar_VCL;

begin
  Toolbar.Theme := utDark;
  StatusBar.Theme := utDark;
  // 0 (the default) uses the accent colour of the theme
  Toolbar.AccentColor := $FF1F883D;
  StatusBar.AccentColor := $FF1F883D;
end;
```

Consulte también

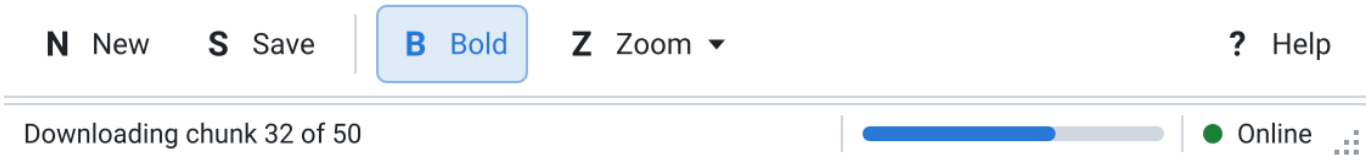
- Progreso e indicadores
- Arquitectura

Renderizado sin interfaz

El renderizador de sgcToolbar_Render pinta un TsgcToolbarModel o un TsgcStatusBarModel sin ventana y sin VCL ni FMX, de modo que una aplicación de consola, un servicio, una prueba o un servidor web pueden producir las mismas barras como imagen, por ejemplo para la documentación de una aplicación o una prueba visual. Las unidades compilan en todas las plataformas, Linux64 incluida.

Rutina	Descripción
Model.SaveToPNG(aFileName or aStream, aWidth, aHeight, aScale = 1)	Escribe un archivo o flujo PNG. aWidth y aHeight son píxeles lógicos y aScale los multiplica: 560 por 44 a escala 2 da una imagen de 1120 por 88.
Model.SaveToSVG(aFileName or aStream, aWidth, aHeight, aScale = 1)	Escribe un archivo o flujo SVG de la misma barra.
Model.Render(aCanvas, aWidth, aHeight, aScale = 1)	Pinta la barra en cualquier lienzo de sgcUICore, en píxeles de dispositivo.
RenderToolbar(aModel, aCanvas, aWidth, aHeight, aScale, aImages)	El renderizador de sgcToolbar_Render. aImages devuelve la imagen de un ImageIndex, nil dibuja el Glyph.
RenderStatusBar(aModel, aCanvas, aWidth, aHeight, aScale)	El renderizador de la barra de estado.

Una barra de herramientas mide de alto ButtonSize más dos veces Padding, 44 píxeles lógicos con los valores por defecto. El modelo de un control está disponible mediante su propiedad **Model**, por lo que **Toolbar.Model.SaveToPNG(...)** guarda lo que muestra el control con cualquier tamaño.



Ejemplo

```
uses
  SysUtils,
  sgcUICore_Theme, sgcToolbar_Model;

var
  oToolbar: TsgcToolbarModel;
  oStatus: TsgcStatusBarModel;
  oItem: TsgcToolbarItem;
  oPanel: TsgcStatusBarPanel;
begin
  oToolbar := TsgcToolbarModel.Create;
  oStatus := TsgcStatusBarModel.Create;
  try
    oItem := oToolbar.Items.Add;
    oItem.Caption := 'New';
    oItem.Glyph := 'N';
    oItem := oToolbar.Items.Add;
    oItem.Caption := 'Save';
    oItem.Glyph := 'S';
    oToolbar.Items.Add.Kind := tikSeparator;
    oItem := oToolbar.Items.Add;
    oItem.Kind := tikToggle;
```

```
    oItem.Caption := 'Bold';
    oItem.Glyph := 'B';
    oItem.Down := True;
    oItem := oToolbar.Items.Add;
    oItem.Kind := tikDropdown;
    oItem.Caption := 'Zoom';
    oItem.Glyph := 'Z';
    oToolbar.Items.Add.Kind := tikSpacer;
    oItem := oToolbar.Items.Add;
    oItem.Caption := 'Help';
    oItem.Glyph := '?';

    oPanel := oStatus.Panels.Add;
    oPanel.Spring := True;
    oPanel.Text := 'Downloading chunk 32 of 50';
    oPanel := oStatus.Panels.Add;
    oPanel.Kind := spkProgress;
    oPanel.Width := 140;
    oPanel.Progress := 64;
    oPanel := oStatus.Panels.Add;
    oPanel.Kind := spkIndicator;
    oPanel.AutoSize := True;
    oPanel.Text := 'Online';
    oPanel.IndicatorState := sisOk;

    // 560 x 44 logical pixels (ButtonSize 32 + 2 x Padding 6),
    // written at scale 2 as 1120 x 88 pixels
    oToolbar.SaveToPNG('toolbar.png', 560, 44, 2);
    oToolbar.SaveToSVG('toolbar.svg', 560, 44);
    oStatus.SaveToPNG('statusbar.png', 560, 24, 2);
    oStatus.SaveToSVG('statusbar.svg', 560, 24);
  finally
    oStatus.Free;
    oToolbar.Free;
  end;
end;
```

Consulte también

- TsgcToolbarModel
- TsgcStatusBarModel
- Plataformas

Descripción general de .NET

sgcToolbar .NET es un port administrado a C# de la biblioteca sgcToolbar de Delphi, no un contenedor alrededor de ella. Las clases de C# están en el espacio de nombres **esegece.sgcToolbar** y conservan los nombres de Delphi, por lo que **TsgcToolbarModel**, **TsgcToolbarItem**, **TsgcToolbarItemKind**, **TsgcStatusBarModel** y **TsgcStatusBarPanel** se leen igual en los dos lenguajes, y la barra de herramientas y la barra de estado de cada host siguen el patrón de nombres de Delphi. Se dibujan con el port .NET de **sgcUICore**, con el mismo lienzo, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, de modo que una barra tiene el mismo aspecto en WinForms, WPF, Avalonia y en la salida sin interfaz en PNG y SVG.
Una barra de herramientas contiene botones, interruptores, grupos de opciones, botones desplegables, separadores, espaciadores y etiquetas, y mueve los elementos que no caben a un menú de desbordamiento. Una barra de estado contiene paneles de texto, de progreso y de indicador con ancho fijo, automático o elástico, y sus paneles se pueden actualizar desde cualquier hilo.

Paquetes

El port se divide en cinco paquetes NuGet, de modo que un proyecto solo incorpora el host que utiliza.

PackageId	Contiene	Frameworks de destino
esegece.sgcToolbar	Los modelos (**TsgcToolbarModel**, **TsgcStatusBarModel**), sus elementos y paneles, el vínculo **Command** del elemento, el renderizador **sgcToolbar_Render** y la salida SVG sin interfaz. Sin dependencia nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcToolbar.Rendering	La salida PNG sin interfaz, los métodos de extensión **SaveToPNG** de **sgcToolbar_Raster**, dibujados en el lienzo SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcToolbar.WinForms	**TsgcWinFormsToolbar** y **TsgcWinFormsStatusBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcToolbar.Wpf	**TsgcWpfToolbar** y **TsgcWpfStatusBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcToolbar.Avalonia	**TsgcAvaloniaToolbar** y **TsgcAvaloniaStatusBar**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0

Los tres paquetes de host dependen de esegece.sgcToolbar.Rendering, que depende de esegece.sgcToolbar, de modo que una sola línea `PackageReference` incorpora todo lo que un host necesita. Consulte Instalación desde NuGet.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcToolbar, esegece.sgcToolbar.Rendering, esegece.sgcToolbar.WinForms, esegece.sgcToolbar.Wpf, esegece.sgcToolbar.Avalonia	Todo: las barras de herramientas y de estado de los tres hosts, todos los tipos de elemento y de panel, grupos de opciones, menús desplegables, el menú de desbordamiento, el vínculo Command del elemento, imágenes, sugerencias, navegación con el teclado, los setters seguros entre hilos de la barra de estado, el tirador de tamaño, los temas y la salida sin interfaz en PNG y SVG.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcToolbar.WinForms.Community. Dependen de los paquetes Community de sgcUICore.	El mismo contenido que Registered sin el menú de desbordamiento y sin las propiedades **Command** y **CommandParameter** del elemento, como la edición Community de Delphi. Los elementos que no caben se recortan en el borde derecho, y los elementos ejecutan su OnClick.

La edición Community es gratuita para uso comercial y no comercial por parte de individuos y organizaciones con menos de EUR 1.000.000 de ingresos brutos anuales y no más de cinco desarrolladores. Una organización que alcance o supere cualquiera de los dos umbrales necesita una licencia Registered. El port .NET no tiene edición Trial.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (TsgcVCLToolbar, TsgcVCLStatusBar)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	FMX (TsgcFMXToolbar, TsgcFMXStatusBar)	Windows, .NET Framework 4.7.2 y .NET 8 y 9. Las barras se declaran en XAML y los elementos y paneles se añaden en código.
Avalonia	FMX (TsgcFMXToolbar, TsgcFMXStatusBar)	Windows, Linux y macOS, .NET 8 y 9, con ratón, táctil y lápiz. Las barras se declaran en XAML y los elementos y paneles se añaden en código.
Headless	Los métodos Render y SaveToSVG de los modelos	Cualquier plataforma de los frameworks de destino: SVG con esegece.sgcToolbar, PNG con esegece.sgcToolbar.Rendering.

Toda barra de herramientas tiene la misma API: Items, ShowCaptions, ButtonSize, Spacing, Padding, Theme, AccentColor, Images, OverflowMenu, ItemAtPos, ItemRect, OverflowRect, ClickItem y OnItemClick. Toda barra de estado tiene Panels, SizeGrip, Theme, AccentColor, PanelAtPos, PanelRect, GripRect, los métodos seguros entre hilos SetPanelText, SetProgress y SetIndicator, y OnPanelClick. Consulte Clases de .NET para los nombres de las clases y Diferencias de .NET con Delphi para lo que cambia respecto a Delphi.

Consulte también

- Instalación desde NuGet
- Inicio rápido de WinForms
- Inicio rápido de WPF
- Inicio rápido de Avalonia

- Clases de .NET
- Diferencias de .NET con Delphi
- Renderizado sin interfaz en .NET
- Demos de .NET
- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, y los paquetes core y de renderizado se incorporan como dependencias.

Edición Registered

```
dotnet add package esegece.sgcToolbar.WinForms
dotnet add package esegece.sgcToolbar.Wpf
dotnet add package esegece.sgcToolbar.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcToolbar.WinForms.Community
dotnet add package esegece.sgcToolbar.Wpf.Community
dotnet add package esegece.sgcToolbar.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcToolbar.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los Community, nunca ambos: contienen los mismos tipos en el mismo espacio de nombres. El código que usa el **Command** del elemento o la propiedad **OverflowMenu** de una barra de herramientas no compila con los paquetes Community; protéjalo con `#if !SGC_COMMUNITY` cuando un proyecto compila las dos ediciones.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto de .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto de .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:sgc="clr-namespace:esegece.sgcToolbar;assembly=esegece.sgcToolbar.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Asigne el espacio de nombres en XAML con `xmlns:sgc="using:esegece.sgcToolbar"`.
- Headless: esegece.sgcToolbar por sí solo basta para SVG. Añada esegece.sgcToolbar.Rendering para PNG, incorpora SkiaSharp y sus recursos nativos para Windows y Linux.

En C# todo host necesita `using esegece.sgcToolbar;`, y `using esegece.sgcUICore;` para **TsgcUITheme** y los demás tipos de los temas.

Consulte también

- Descripción general de .NET
- Inicio rápido de WinForms
- Inicio rápido de WPF
- Inicio rápido de Avalonia
- Demos de .NET

Inicio rápido de WinForms

TsgcWinFormsToolbar y **TsgcWinFormsStatusBar** son las barras de WinForms, los equivalentes de TsgcVCLToolbar y TsgcVCLStatusBar de Delphi. Cada una posee un modelo, expone sus propiedades y lo dibuja mediante sgcUICore. El paquete es **esegece.sgcToolbar.WinForms**.

Pasos

- Cree un proyecto WinForms y añada el paquete esegece.sgcToolbar.WinForms.
- Cree un **TsgcWinFormsToolbar**. Se acopla arriba por defecto y su altura sigue a **ButtonSize** y **Padding**.
- Añada los elementos con `Items.Add()` y establezca su **Kind**, **Caption**, **Glyph** o **ImageIndex**, y **Hint**. Un **tikToggle** con el mismo **GroupIndex** forma un grupo de opciones, y un **tikDropdown** muestra el **ContextMenuStrip** de su **DropdownMenu**.
- Cree un **TsgcWinFormsStatusBar**, que se acopla abajo, y añada sus paneles con `Panels.Add()`.
- Asigne **OnItemClick** para reaccionar a cada clic, o **OnClick** de cada elemento.

Ejemplo

El Program.cs completo de un editor WinForms con una barra de herramientas y una barra de estado: dos botones, un interruptor Bold, un grupo de opciones Left y Center, un desplegable Zoom, y un panel de texto elástico, un panel de progreso y un indicador. Compila tal cual en un proyecto `net8.0-windows` que referencia esegece.sgcToolbar.WinForms.
Las barras también están en la caja de herramientas. El diseñador no serializa los elementos ni los paneles, por lo que siempre se añaden en código, consulte Diferencias de .NET con Delphi. Establezca **TabStop** en True para mover el foco entre los botones con las teclas de flecha.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET con Delphi
- Inicio rápido de WPF
- Inicio rápido de Avalonia
- Tipos de elemento

Inicio rápido de WPF

TsgcWpfToolbar y **TsgcWpfStatusBar** son las barras de WPF. Se declaran en XAML como cualquier control, y sus elementos y paneles se añaden en código. El paquete es **esegece.sgcToolbar.Wpf**.

Pasos

- Cree un proyecto WPF (`dotnet new wpf`) y añada el paquete esegece.sgcToolbar.Wpf.
- Asigne el espacio de nombres `esegece.sgcToolbar` y declare un **TsgcWpfToolbar** y un **TsgcWpfStatusBar**. **ShowCaptions**, **ButtonSize**, **SizeGrip** y las demás propiedades simples se pueden establecer como atributos XAML.
- En el code-behind asigne **Images**, una lista de **ImageSource**, añada los elementos y los paneles y asigne **OnItemClick**: es un delegado, no un evento de WPF, por lo que no se puede establecer como atributo XAML.

Ejemplo

MainWindow.xaml declara la barra de herramientas, la barra de estado y un cuadro de texto. El App.xaml del proyecto es el de la plantilla de WPF, con StartupUri establecido en MainWindow.xaml.
MainWindow.xaml.cs crea tres iconos vectoriales **DrawingImage**, añade los botones, un interruptor Wrap y un desplegable Zoom con un **ContextMenu** de WPF, y un panel de texto y un indicador.
Images acepta cualquier **ImageSource**, tanto un **BitmapImage** como un **DrawingImage**. Cuando la lista es una **ObservableCollection<ImageSource>**, la barra de herramientas se redibuja cuando se añade, elimina o reemplaza una imagen.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET con Delphi
- Inicio rápido de WinForms
- Inicio rápido de Avalonia
- Imágenes y glifos

Inicio rápido de Avalonia

TsgcAvaloniaToolbar y **TsgcAvaloniaStatusBar** son las barras de Avalonia, los equivalentes de TsgcFMXToolbar y TsgcFMXStatusBar de Delphi, en Windows, Linux y macOS. El ratón, el táctil y el lápiz pasan por la misma ruta de puntero. El paquete es **esegece.sgcToolbar.Avalonia**.

Pasos

- Cree un proyecto .NET 8 y añada los paquetes esegece.sgcToolbar.Avalonia, Avalonia.Desktop y Avalonia.Themes.Fluent.
- Asigne el espacio de nombres con `xmlns:sgc="using:esegece.sgcToolbar"` y declare un **TsgcAvaloniaToolbar** y un **TsgcAvaloniaStatusBar**.
- En el code-behind establezca el **Theme**, añada los elementos y los paneles y asigne **OnItemClick**: es un delegado, no un evento de Avalonia.

Ejemplo

MainWindow.axaml declara las dos barras y un texto.
MainWindow.axaml.cs establece el tema oscuro y añade un grupo de opciones Grid y List, un espaciador que empuja los elementos siguientes hacia la derecha, una etiqueta, un botón Refresh, un panel de texto elástico y un panel de progreso.
Program.cs inicia la aplicación con el tema Fluent.
Theme oculta la propiedad `StyledElement.Theme` de los controles de Avalonia, así que establézcala en código en una variable del tipo de la barra, no como atributo XAML. Images es una lista de **IImage**, un **Bitmap** o un **DrawingImage**.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET con Delphi
- Inicio rápido de WinForms
- Inicio rápido de WPF
- Interruptores y grupos de opciones

Clases de .NET

Cada host tiene una clase de barra de herramientas y una clase de barra de estado con los mismos miembros. Los tipos independientes del framework son compartidos por los tres hosts y conservan sus nombres de Delphi.

Clases de host

Host	Clase	Clase base	Delphi
WinForms	TsgcWinFormsToolbar	TsgcUIWinFormsControl	TsgcVCLToolbar
WinForms	TsgcWinFormsStatusBar	TsgcUIWinFormsControl	TsgcVCLStatusBar
WPF	TsgcWpfToolbar	TsgcUIWpfControl	TsgcFMXToolbar
WPF	TsgcWpfStatusBar	TsgcUIWpfControl	TsgcFMXStatusBar
Avalonia	TsgcAvaloniaToolbar	TsgcUIAvaloniaControl	TsgcFMXToolbar
Avalonia	TsgcAvaloniaStatusBar	TsgcUIAvaloniaControl	TsgcFMXStatusBar

Tipos independientes del framework

Tipo	Descripción
TsgcToolbarModel	Los elementos, el diseño con la lista de desbordamiento, los elementos resaltados, pulsados y con foco y la prueba de impacto. Toda barra de herramientas posee uno, consulte TsgcToolbarModel.
TsgcToolbarItems	La colección de elementos: Count, el indexador, Add, Insert, Delete, Clear, IndexOf, BeginUpdate y EndUpdate.
TsgcToolbarItem	Un elemento: Kind, Caption, Glyph, ImageIndex, Hint, Enabled, Visible, Down, GroupIndex, Width, Tag, DropdownMenu, OnClick, Click(), los resultados del diseño BoundsRect, Shown e InOverflow, y en la edición Registered Command y CommandParameter. Consulte TsgcToolbarItem.
TsgcToolbarItemKind	tikButton, tikToggle, tikDropdown, tikSeparator, tikSpacer, tikLabel.
TsgcStatusBarModel	Los paneles, su diseño, el tirador de tamaño y los setters seguros entre hilos. Toda barra de estado posee uno, consulte TsgcStatusBarModel.
TsgcStatusBarPanels	La colección de paneles: Count, el indexador, Add, Insert, Delete, Clear, IndexOf, BeginUpdate y EndUpdate.
TsgcStatusBarPanel	Un panel: Kind, Text, Width, AutoSize, Spring, Alignment, Progress, IndicatorState, Hint, Tag y BoundsRect. Consulte TsgcStatusBarPanel.
TsgcStatusPanelKind	spkText, spkProgress, spkIndicator.
TsgcStatusIndicatorState	sisOff, sisOk, sisWarning, sisError.

TAlignment	taLeftJustify, taRightJustify, taCenter: la alineación del texto de un panel, como el tipo de la RTL de Delphi.
TsgcToolbarItemEvent	`void (object Sender, TsgcToolbarItem aItem)` , el tipo de OnItemClick.
TsgcStatusBarPanelEvent	`void (object Sender, TsgcStatusBarPanel aPanel)`, el tipo de OnPanelClick.
sgcToolbar_Render, sgcToolbar_Raster	El renderizador y la salida PNG sin interfaz, consulte Renderizado sin interfaz en .NET.

Barra de herramientas

Miembro	Tipo	Descripción
Model	TsgcToolbarModel	El modelo de la barra de herramientas.
Items	TsgcToolbarItems	Los elementos, en orden de izquierda a derecha.
ShowCaptions	bool	Dibuja el título de los botones junto a su icono. Sin ella, un botón con icono muestra solo el icono.
ButtonSize, Spacing, Padding	int	La altura de un botón, el espacio entre elementos y el relleno a su alrededor, 32, 4 y 6 por defecto. La altura de la barra de herramientas sigue a ButtonSize y Padding.
Theme, AccentColor	TsgcUITheme, uint	El tema de la barra. AccentColor 0 usa el acento del tema.
Images	ImageList, IList<ImageSource>, IList<IImage>	Las imágenes de ImageIndex, consulte Diferencias de .NET con Delphi.
OverflowMenu	ContextMenuStrip, ContextMenu	El menú del botón de desbordamiento, reconstruido cada vez que se abre. Solo en la edición Registered.
TabStop (WinForms), Focusable (WPF, Avalonia)	bool	Permite que la barra reciba el foco: Left, Right, Home y End mueven el botón con foco, Enter y Space lo activan.
ItemAtPos(X, Y)	int	El índice del elemento en un punto del control, -1, o CS_TOOLBAR_HIT_OVERFLOW sobre el botón de desbordamiento.
ItemRect(aIndex), OverflowRect()	Rectangle, Rect	El rectángulo de un elemento o del botón de desbordamiento en las coordenadas del control.
ClickItem(aIndex)	void	Activa un elemento como lo hace un clic.
OnItemClick	TsgcToolbarItemEvent	Un clic, Enter o Space en un botón habilitado, después de que un interruptor cambie Down y antes de que se ejecute el OnClick o el Command del elemento.

Barra de estado

Miembro	Tipo	Descripción

Model	TsgcStatusBarModel	El modelo de la barra de estado.
Panels	TsgcStatusBarPanels	Los paneles, en orden de izquierda a derecha.
SizeGrip	bool	Dibuja el tirador de tamaño en el extremo derecho y cambia el tamaño de la ventana cuando se arrastra, consulte Diferencias de .NET con Delphi.
Theme, AccentColor	TsgcUITheme, uint	El tema de la barra. AccentColor 0 usa el acento del tema.
SetPanelText(aIndex, aText), SetProgress(aIndex, aValue), SetIndicator(aIndex, aState)	void	Establecen el Text, el Progress (0 a 100) y el IndicatorState de un panel desde cualquier hilo. Un índice fuera de rango se ignora.
PanelAtPos(X, Y)	int	El índice del panel en un punto del control, o -1.
PanelRect(aIndex), GripRect()	Rectangle, Rect	El rectángulo de un panel o del tirador de tamaño en las coordenadas del control, vacío sin SizeGrip.
OnPanelClick	TsgcStatusBarPanelEvent	Un clic en un panel.
CS_STATUSBAR_HEIGHT	const int	La altura por defecto, 24.

Las posiciones y los tamaños son píxeles en WinForms (Rectangle) y unidades independientes del dispositivo en WPF y Avalonia (Rect).

Consulte también

- Descripción general de .NET
- Diferencias de .NET con Delphi
- Inicio rápido de WinForms
- Inicio rápido de WPF
- Inicio rápido de Avalonia

Diferencias de .NET con Delphi

La API de .NET sigue a la API de Delphi nombre por nombre. Estos son los puntos en los que difiere.

Command en lugar de vínculos de acción

Delphi vincula un elemento a una TBasicAction mediante su propiedad Action, consulte Vínculos de acción. .NET no tiene TBasicAction, así que el elemento tiene **Command**, un `System.Windows.Input.ICommand`, y **CommandParameter**. Ambos forman parte únicamente de la edición Registered, como los vínculos de acción de Delphi.

- `CanExecute(CommandParameter)` controla el **Enabled** del elemento: cuando se asigna el comando y en cada `CanExecuteChanged`.
- Un clic, Enter o Space cambia un interruptor, dispara OnItemClick de la barra y después ejecuta `Click()` del elemento: su manejador OnClick cuando hay uno asignado, y si no el comando, cuando puede ejecutarse.
- Caption, Hint y Down no se sincronizan desde el comando, porque ICommand no lleva ninguno. Siguen siendo propiedades del elemento, de modo que un interruptor conserva su propio estado Down. ImageIndex y Visible tampoco se sincronizan.
- No hay OnUpdate en reposo. Un comando que cambia de estado dispara CanExecuteChanged, y el elemento lo sigue de inmediato.

Un programa de consola que vincula un elemento Save a un comando mínimo. El elemento empieza deshabilitado, se habilita cuando cambia el documento, y un clic guarda y lo deshabilita de nuevo. Solo necesita esegece.sgcToolbar, las barras de herramientas de los tres hosts se comportan igual.

Menús desplegables y de desbordamiento

En Delphi **DropdownMenu** es un TPopupMenu de VCL o de FireMonkey, consulte Menús desplegables. En .NET se declara como `object`, para que el modelo siga siendo independiente del framework, y cada host muestra bajo el botón el tipo de menú de su framework. **OverflowMenu** lo construye la barra de herramientas cada vez que se abre, con una entrada por elemento de la lista de desbordamiento, y elegir una entrada activa el elemento como lo hace un clic, consulte Desbordamiento.

Host	DropdownMenu	OverflowMenu
WinForms	System.Windows.Forms.ContextMenuStrip. Un menú liberado se borra de su elemento.	ContextMenuStrip, el Hint de un elemento es el ToolTipText de su entrada.
WPF	System.Windows.Controls.ContextMenu.	ContextMenu, con un Separator para un elemento separador y el Hint como ToolTip de una entrada.
Avalonia	Avalonia.Controls.ContextMenu.	ContextMenu, con un Separator para un elemento separador y el Hint como sugerencia de una entrada.

El menú se abre cuando se pulsa el botón, después de OnItemClick y del OnClick del elemento, de modo que un manejador puede rellenar o marcar el menú antes de que se muestre. El menú de desbordamiento no forma parte de la edición Community: allí los elementos que no caben se recortan y la propiedad OverflowMenu no existe.

Imágenes

Delphi usa una TCustomImageList en VCL y una TCustomImageList de FMX.ImgList en FireMonkey, consulte Imágenes y glifos. En .NET la propiedad **Images** de la barra de herramientas tiene el tipo de imagen de cada host, y el **ImageIndex** del elemento es un índice en ella.

Host	Images	Se dibuja a

Win-Forms	System.Windows.Forms.ImageList	El ImageSize de la lista de imágenes, en píxeles.
WPF	IList<ImageSource>: mapas de bits e iconos vectoriales DrawingImage. Los iconos vectoriales se rasterizan con Skia, por lo que también se dibujan en una sesión de Escritorio remoto minimizada o desconectada. Una ObservableCollection se redibuja cuando cambia.	16 x 16 unidades independientes del dispositivo, nítida con cualquier DPI para los iconos vectoriales.
Avalonia	IList<IImage>: un Bitmap o un DrawingImage. Una lista con INotifyCollectionChanged se redibuja cuando cambia.	16 x 16 unidades independientes del dispositivo.

Cada imagen se convierte una vez a ARGB y se guarda en caché hasta que cambia la lista, y en WPF y Avalonia también hasta que cambia la escala de DPI. Cuando un elemento no tiene imagen, el texto de su **Glyph** se dibuja en el hueco del icono.

Sugerencias y tooltips

En Delphi las barras muestran el Hint del elemento o del panel bajo el cursor cuando ShowHint es True, y FMX muestra sugerencias desde 10.1 Berlin. En .NET las barras siempre muestran un tooltip, sin propiedad ShowHint: el Hint del elemento bajo el cursor, o su Caption cuando el título no se dibuja, y en la barra de estado el Hint del panel. WinForms usa un componente ToolTip interno, WPF establece la propiedad ToolTip de la barra y Avalonia establece la propiedad adjunta ToolTip.Tip. No establezca usted mismo el tooltip de una barra, la barra lo reemplaza; establezca en su lugar el Hint del elemento o del panel.

Actualizaciones de estado seguras entre hilos

Como en Delphi, **SetPanelText**, **SetProgress** y **SetIndicator** de la barra de estado pueden llamarse desde cualquier hilo, consulte Actualizaciones seguras entre hilos. El modelo protege los valores con un bloqueo y dispara su OnChange desde el hilo que llama, solo cuando el valor cambió. La barra responde con una solicitud de redibujado y, desde un hilo de trabajo, nunca toca el control:

- WinForms y WPF envían una devolución de llamada mediante el `SynchronizationContext` que el control capturó en el hilo de la interfaz.
- Avalonia envía una devolución de llamada a `Dispatcher.UIThread`.
- Solo hay una devolución de llamada pendiente a la vez, de modo que un hilo de trabajo que actualiza la barra miles de veces por segundo no inunda la cola de la interfaz. La devolución de llamada redibuja con los últimos valores.

No hace falta ninguna llamada a Invoke, BeginInvoke o Dispatcher. Todas las demás propiedades, y añadir, eliminar o configurar paneles, deben hacerse desde el hilo de la interfaz. El formulario siguiente descarga 50 fragmentos simulados en un hilo de trabajo y detiene el hilo antes de que se cierre el formulario.

Tirador de tamaño

Con **SizeGrip** en True, el valor por defecto, la barra de estado dibuja los puntos del tirador en su extremo derecho y **GripRect** devuelve su rectángulo, consulte Tirador de tamaño. En Delphi solo la barra VCL cambia el tamaño del formulario; la barra FMX solo dibuja el tirador. En .NET el tirador cambia el tamaño de la ventana en todos los hosts, cuando la ventana admite cambio de tamaño y no está maximizada, y nunca en tiempo de diseño:

Host	Cómo el tirador cambia el tamaño de la ventana
Win-Forms	La barra responde a la prueba de impacto del área no cliente del tirador con HTBOTTOMRIGHT, como la barra VCL, para un formulario con borde redimensionable.
WPF	Una pulsación en el tirador inicia el bucle de cambio de tamaño del sistema para la ventana (WM_SYSCOMMAND con SC_SIZE), para una Window con ResizeMode CanResize o CanResizeWithGrip. El cursor cambia sobre el tirador.

Avalonia	Una pulsación en el tirador llama a BeginResizeDrag de la ventana, para una Window con CanResize, en Windows, Linux y macOS.

Unidades

- WinForms: ButtonSize, Spacing, Padding y el Width de elementos y paneles son píxeles lógicos multiplicados por DPI / 96, como las barras VCL con PPI / 96. ItemAtPos, ItemRect, PanelRect y GripRect son píxeles de cliente.
- WPF y Avalonia: ItemAtPos, ItemRect, OverflowRect, PanelAtPos, PanelRect y GripRect son unidades independientes del dispositivo, como en FMX. El marco y el diseño de los modelos son píxeles de dispositivo, las unidades multiplicadas por la escala de DPI del control (WPF) o la escala de renderizado de la ventana (Avalonia). El BoundsRect de un elemento o de un panel también está en píxeles de dispositivo.
- Headless: aWidth y aHeight son píxeles lógicos y aScale los multiplica, consulte Renderizado sin interfaz en .NET.

Eventos

OnItemClick, OnPanelClick y el OnClick del elemento son propiedades delegado con los nombres de Delphi, no miembros `event` de C#. Asignar un manejador reemplaza al anterior, como un puntero a método de Delphi, y se asignan en código, también en WPF y Avalonia, donde XAML no puede establecerlos.

Sin verbos del diseñador

En Delphi las propiedades Items y Panels se editan en tiempo de diseño con el editor de colecciones estándar. El port .NET no tiene verbos del diseñador ni editor de colecciones: los Items y Panels de una barra no se serializan, por lo que siempre se añaden en código, normalmente en el constructor del formulario o de la ventana. En WinForms las barras están en la caja de herramientas y sus propiedades simples se establecen en la ventana Propiedades. En WPF y Avalonia las barras se declaran en XAML.

Ediciones

El port .NET se distribuye en paquetes Registered y Community, consulte Descripción general de .NET. Los paquetes Community no tienen OverflowMenu, Command ni CommandParameter, como la edición Community de Delphi, consulte Ediciones. No hay edición Trial.

Enlace de datos

En WPF y Avalonia las propiedades visibles para el usuario son propiedades de dependencia y propiedades de Avalonia, por lo que se pueden enlazar, aplicar con estilos y animar, consulte Enlace de datos en .NET.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Inicio rápido de WinForms
- Inicio rápido de WPF
- Inicio rápido de Avalonia
- Vínculos de acción
- Menús desplegables
- Imágenes y glifos
- Actualizaciones seguras entre hilos
- Tirador de tamaño

Enlace de datos en .NET

En los hosts WPF y Avalonia, las propiedades visibles para el usuario de la barra de herramientas y de la barra de estado son propiedades de dependencia (WPF) y propiedades con estilo de Avalonia, por lo que se pueden enlazar a un modelo de vista, establecer mediante un Style y animar. Este tema se aplica solo a WPF y Avalonia. Los controles WinForms no cambian: sus propiedades son propiedades CLR simples, como antes.

Propiedades enlazables

Cada propiedad de la tabla tiene un campo estático con el sufijo Property, por ejemplo `TsgcWpfToolbar.ShowCaptionsProperty`. En WPF los campos son DependencyProperty y en Avalonia son StyledProperty. El usuario no modifica ninguna de estas propiedades, por lo que todas se enlazan OneWay de forma predeterminada. Utilice `Mode=TwoWay` cuando el modelo de vista deba recibir también un valor establecido en el control desde el código.

Clase WPF	Clase Avalonia	Propiedades enlazables
TsgcWpfToolbar	TsgcAvaloniaToolbar	ShowCaptions, ButtonSize, Spacing, Padding, Theme, AccentColor
TsgcWpfStatusBar	TsgcAvaloniaStatusBar	Theme, SizeGrip, AccentColor

Padding es el relleno de la barra en unidades independientes del dispositivo, un int. En WPF oculta el Padding de tipo Thickness de Control. En Avalonia, Theme es el tema de sgcUICore (TsgcUITheme) y oculta el tema de control de Avalonia de StyledElement, que sigue accesible mediante `StyledElement.ThemeProperty`.

Comportamiento

- **El modelo conserva los valores.** Cada propiedad se guarda en el modelo de la barra de herramientas o de la barra de estado. Un valor enlazado pasa al modelo, y la propiedad recupera el valor que conserva el modelo: el modelo eleva ButtonSize a 8 y Spacing y Padding a 0, y la propiedad muestra entonces el valor en uso.
- **Un valor corregido llega al origen.** Un origen TwoWay también recibe el valor corregido, en WPF y en Avalonia. WPF no actualiza el origen mientras transfiere un valor desde él, por lo que en WPF el control escribe de vuelta el valor corregido cuando el dispatcher ha ejecutado el trabajo pendiente. Un cambio posterior hecho en el control, desde el código o en el modelo, también llega a un origen TwoWay en ambas plataformas. En Avalonia la escritura de vuelta cubre un valor que el origen envía cuando el enlace ya está en marcha. Un valor que el origen ya tiene cuando el enlace se inicia se corrige en el control, pero no se escribe de vuelta en el origen.
- **Los cambios hechos en el modelo se copian de vuelta.** Cuando el modelo cambia un valor por sí mismo, el control lo copia en la propiedad, de modo que los enlaces y los estilos ven el valor en uso.
- **Styles.** Un setter de Style puede establecer cualquier propiedad enlazable, por ejemplo el Theme de todas las barras de herramientas y barras de estado de una ventana.

Propiedades que siguen siendo propiedades CLR

Propiedad	Motivo
Items (barra de herramientas)	La colección de botones del modelo, construida en código, no un valor único que enlazar.
Images (barra de herramientas)	La lista de imágenes de los botones, un objeto compartido por referencia.

OverflowMenu (barra de herramientas)	El menú que la barra de herramientas construye para los botones que no caben, de solo lectura.
Panels (barra de estado)	La colección de paneles del modelo. Los valores de los paneles (Text, Progress, IndicatorState) se establecen mediante los métodos seguros para subprocesos SetPanelText, SetProgress y SetIndicator, que se pueden llamar desde cualquier subproceso.
Model	El modelo independiente de la plataforma que hay detrás de cada control, de solo lectura.

Ejemplo

El modelo de vista siguiente se comparte en ambos ejemplos. ShowCaptions y Theme se enlazan OneWay, y ButtonSize se enlaza con `Mode=TwoWay`, de modo que un tamaño establecido en la barra de herramientas desde el código vuelve al modelo de vista. Los botones se agregan en código, como en los inicios rápidos.

WPF

MainWindow.xaml crea el modelo de vista como DataContext de la ventana. El código subyacente y App.xaml son los de la plantilla WPF.

Avalonia

MainWindow.axaml establece `x:DataType` con el modelo de vista, por lo que los enlaces son enlaces compilados. El código subyacente llama a InitializeComponent, y la clase App es la de Inicio rápido Avalonia.

Consulte también

- Diferencias de .NET respecto a Delphi
- Clases de .NET
- Inicio rápido WPF
- Inicio rápido Avalonia

Renderizado sin interfaz en .NET

Una barra de herramientas o una barra de estado se puede dibujar sin ventana, en un servidor, en una prueba o en un informe. **SaveToSVG** de ambos modelos escribe un archivo o flujo SVG solo con el paquete esegece.sgcToolbar, y los métodos de extensión **SaveToPNG** de esegece.sgcToolbar.Rendering escriben un PNG dibujado en SkiaSharp, en Windows, Linux y macOS. La salida SVG es idéntica byte a byte a la del renderizado sin interfaz de Delphi.

Métodos

Método	Paquete	Descripción
TsgcToolbarModel.SaveToSVG y TsgcStatusBarModel.SaveToSVG(aFileName o aStream, aWidth, aHeight, aScale = 1)	esegece.sgcToolbar	Escribe la barra como SVG.
SaveToPNG(aFileName o aStream, aWidth, aHeight, aScale = 1), un método de extensión de ambos modelos en sgcToolbar_Raster	esegece.sgcToolbar.Rendering	Escribe la barra como PNG sobre fondo transparente.
TsgcToolbarModel.Render y TsgcStatusBarModel.Render(aCanvas, aWidth, aHeight, aScale = 1)	esegece.sgcToolbar	Dibuja la barra en cualquier lienzo de sgcUICore, una barra de aWidth x aHeight píxeles de dispositivo.
sgcToolbar_Render.RenderToolbar(aModel, aCanvas, aWidth, aHeight, aScale, aImages), sgcToolbar_Render.RenderStatusBar(aModel, aCanvas, aWidth, aHeight, aScale)	esegece.sgcToolbar	El renderizador que hay detrás de Render.

Para SaveToSVG y SaveToPNG, aWidth y aHeight son píxeles lógicos. La imagen mide aWidth x aScale por aHeight x aScale píxeles, así que una escala de 2 da una imagen nítida para una pantalla de alta densidad o para imprimir. El modelo de la barra de herramientas no tiene lista de imágenes: su evento **OnGetImage**, `bool (int aIndex, out TsgcUIARGBBuffer aImage)`, devuelve la imagen de un ImageIndex como un **TsgcUIARGBBuffer**; sin él, el Glyph del elemento se dibuja en el hueco del icono.

Ejemplo

Un programa de consola (`net8.0`) que referencia esegece.sgcToolbar.Rendering y escribe una barra de herramientas con un botón deshabilitado, un interruptor pulsado, un desplegable y una etiqueta empujada a la derecha por un espaciador, y una barra de estado oscura con un panel de texto elástico, un panel de progreso en 65 y un indicador de advertencia, cada una en PNG y SVG.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Renderizado sin interfaz
- TsgcToolbarModel

- TsgcStatusBarModel

Demos de .NET

Las demos de .NET están en la carpeta `demos` del producto y en el zip de demos, con la solución `sgcToolbar_Demos.sln`. Cada demo tiene un README.md.

Demo	Host	Descripción
EditorShell	WinForms	La estructura de un pequeño editor de texto con un TsgcWinFormsToolbar y un TsgcWinFormsStatusBar. La barra de herramientas contiene los botones New, Open y Save, los interruptores Bold e Italic, un grupo de opciones Left, Center y Right (GroupIndex 1), un desplegable Zoom con un menú de cuatro niveles de zoom, un espaciador, un interruptor de tema oscuro y un botón Help. La barra de estado muestra la línea y la columna del cursor (AutoSize), un panel elástico con el último mensaje, el zoom y un indicador que se pone amarillo mientras el documento tiene cambios sin guardar. Save solo está habilitado mientras hay cambios sin guardar, y al estrechar la ventana los elementos que no caben pasan al menú de desbordamiento. Los botones usan los manejadores OnClick de los elementos y OnItemClick, de modo que la demo compila con los paquetes Community y Registered.
EditorShellWpf	WPF	El mismo editor sobre un TsgcWpfToolbar y un TsgcWpfStatusBar.
EditorShellAvalonia	Avalonia	El mismo editor sobre un TsgcAvaloniaToolbar y un TsgcAvaloniaStatusBar, en Windows, Linux y macOS.
LiveStatus	WinForms	Una barra de estado en vivo alimentada desde un hilo de trabajo. Start ejecuta un hilo que simula una descarga de 50 fragmentos, sin red, e informa de su progreso directamente a la barra de estado con los métodos seguros entre hilos SetPanelText, SetProgress y SetIndicator. La barra de estado tiene un panel de mensajes elástico, un panel de progreso, un indicador del flujo (verde mientras la descarga está en curso, amarillo mientras un fragmento es lento, rojo cuando está detenida) y un contador de fragmentos AutoSize. Stop finaliza el hilo, que informa de su final al hilo de la interfaz mediante el SynchronizationContext.
LiveStatusWpf	WPF	La misma barra de estado en vivo sobre un TsgcWpfStatusBar.
LiveStatusAvalonia	Avalonia	La misma barra de estado en vivo sobre un TsgcAvaloniaStatusBar.

Compilar y ejecutar

Las demos necesitan el SDK de .NET 8 o posterior. Ejecute una demo desde su carpeta:

```
dotnet run
```

Dentro del árbol del producto las demos compilan contra los proyectos de la biblioteca. Desde el zip de demos restauran los paquetes NuGet, por defecto la edición Community. Establezca **SgcToolbarEdition** para usar los paquetes Registered:

```
dotnet build -p:SgcToolbarEdition=Registered
```

Autocomprobación

Cada demo tiene un modificador `--test` que abre la ventana fuera de pantalla y la maneja: EditorShell hace clic en el grupo de opciones, en los interruptores, en Save, en el menú Zoom, en el menú de desbordamiento de una ventana estrecha y en el interruptor Dark, y LiveStatus ejecuta una descarga hasta el final, detiene una segunda a la mitad y cambia el tema. La demo escribe una línea PASS o FAIL por comprobación en el archivo y termina con el número de comprobaciones fallidas, así que 0 significa que todas se superaron.

```
dotnet run -- --test result.txt
```

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido de WinForms
- Inicio rápido de WPF
- Inicio rápido de Avalonia
- Demos

Solución de problemas

Undeclared identifier Action u OverflowMenu

La edición Community compila sin SGC_TOOLBAR_ADVANCED, por lo que la propiedad Action de los elementos y la propiedad OverflowMenu de las barras de herramientas no existen. El código que las usa falla con E2003 Undeclared identifier, y un formulario guardado por una compilación Registered con un elemento vinculado a una acción lanza una excepción "Error reading" al cargarse. Use el OnClick del elemento o el evento OnItemClick de la barra, o una compilación Registered o Trial. Consulte Ediciones.

Los últimos elementos se cortan en el borde derecho

En la edición Community no hay menú de desbordamiento, por lo que los elementos que no caben se recortan. Ensanche la barra, establezca ShowCaptions en False o use una compilación Registered o Trial. Consulte Desbordamiento.

El código posterior a un menú se ejecuta tarde en FireMonkey

El TPopupMenu de FireMonkey de un elemento desplegable y del botón de desbordamiento es modal: Popup no regresa hasta que el menú se cierra. El código colocado después de ClickItem en un elemento desplegable, o en OnItemClick después de mostrar el menú, se ejecuta solo cuando el usuario ha cerrado el menú. Haga el trabajo en el OnClick de las entradas del menú y no llame a ClickItem en un elemento desplegable desde un temporizador ni desde una prueba que no deba esperar. En VCL el menú se abre al pulsar el botón y su bucle modal se lleva el evento de soltar el ratón.

La barra de herramientas ignora la Height que establezco

La altura de una barra de herramientas siempre sigue a ButtonSize más dos veces Padding. Cambie esas dos propiedades en su lugar.

Un interruptor de opción no se puede desactivar

Un interruptor con GroupIndex mayor que 0 es un botón de opción: un clic solo lo activa. Establezca Down en False desde código, o use GroupIndex 0 para un interruptor independiente. Consulte Interruptores y grupos de opciones.

Un elemento no reacciona a un clic

Un elemento deshabilitado (Enabled False) se dibuja atenuado e ignora los clics y el teclado, y los separadores, espaciadores y etiquetas nunca reciben clics. Un elemento con una Action sigue el Enabled de la acción, por lo que compruebe también el OnUpdate de la acción.

La barra de estado parpadea o lanza una excepción al actualizarse desde un hilo

Solo SetPanelText, SetProgress y SetIndicator son seguros entre hilos. Establecer otras propiedades, o añadir paneles, desde un hilo de trabajo toca el control fuera del hilo principal. Consulte Actualizaciones seguras entre hilos.

El tirador de tamaño no redimensiona el formulario

En FMX el tirador solo se dibuja. En VCL funciona únicamente cuando el formulario tiene un borde redimensionable y no está maximizado. Consulte Tirador de tamaño.

Las imágenes o los hints no se muestran en FireMonkey

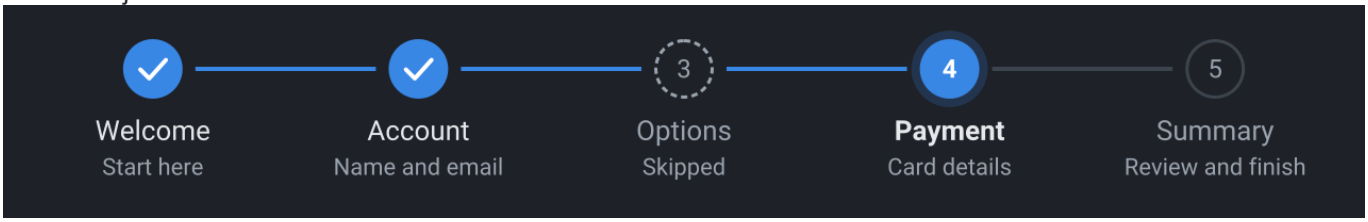
La propiedad Images de la barra de herramientas FMX necesita Delphi XE8 o posterior, y los hints de las barras FMX necesitan 10.1 Berlin o posterior con ShowHint en True. En versiones anteriores los elementos dibujan su Glyph. Consulte Imágenes y glifos.

La demo EditorShell o EditorShellFMX no compila

Ambas demos usan vínculos de acción y el menú de desbordamiento, que no forman parte de la edición Community. Consulte Demos.

Introducción

sgcWizard agrega asistentes paso a paso e indicadores de pasos a las aplicaciones de Delphi y C++Builder, para VCL y FireMonkey. **TsgcVCLWizard** y **TsgcFMXWizard** son contenedores de páginas: cada página es un paso, una banda de stepper muestra dónde está el usuario, y una barra de botones se mueve entre las páginas con Back, Next, Finish y Cancel. **TsgcVCLStepper** y **TsgcFMXStepper** son los indicadores de pasos por sí solos, horizontales o verticales, para cualquier flujo que necesite mostrar su progreso.
sgcWizard forma parte de la suite visual de eSeGeCe, junto a sgcDashboard, sgcNavigation, sgcGrid y sgcToolbar. Se dibuja con **sgcUICore**, el paquete de ejecución compartido con el lienzo, el rasterizador, las fuentes, el escritor SVG y los temas claro y oscuro, de modo que un asistente tiene el mismo aspecto en VCL, en FireMonkey y en la salida PNG y SVG sin interfaz, y coincide con los demás productos de la suite. sgcWizard requiere sgcUICore, que se instala junto con él.



Ediciones

sgcWizard se entrega como Registered, Trial y Community. Todas las ediciones tienen todas las características, solo la compilación Trial muestra un aviso de prueba. Consulte Ediciones.

Por dónde empezar

- Arquitectura: el modelo de stepper independiente del framework, el renderizador sobre el lienzo de sgcUICore y los controles VCL y FMX.
- Plataformas: las versiones de Delphi, C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX e Inicio rápido C++Builder: un primer asistente en pocos minutos.
- Componentes: los asistentes, las páginas, los steppers, los pasos y el modelo.
- Navegación y páginas omitidas, Validación, Finalizar y cancelar, Teclado, Estados de los pasos, Disposición del stepper, Títulos de los botones, Temas, Tiempo de diseño y Renderizado sin interfaz.
- Demos: los proyectos de demostración de Delphi y C++Builder.

Arquitectura

sgcWizard está construido en capas. Cada capa solo depende de las inferiores, de modo que el modelo y el renderizador se compilan sin VCL ni FireMonkey.

Capa	Unidad	Función
Modelo	sgcWizard_Model	TsgcStepperModel contiene los Steps (una colección de TsgcWizardStep), la orientación, el paso activo, los estados automáticos, los tamaños, el tema y el paso bajo el cursor. Su método Layout calcula el círculo y la etiqueta de cada paso visible y los conectores entre ellos, y HitTest encuentra el paso situado bajo un punto.
Renderizador	sgcWizard_Render	sgcWizardRender dibuja un modelo en cualquier lienzo de sgcUICore y sgcWizardRenderButtonBar dibuja la barra de botones. sgcWizardSaveToPNG y sgcWizardSaveToSVG los usan para escribir una imagen sin ventana. Consulte Renderizado sin interfaz.
Controles VCL	sgcWizard_VCL	TsgcVCLWizard, TsgcVCLWizardPage y TsgcVCLStepper. Solo para Windows.
Controles FMX	sgcWizard_FMX	TsgcFMXWizard, TsgcFMXWizardPage y TsgcFMXStepper, con la misma API y el mismo comportamiento.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme	El entorno de ejecución compartido: el lienzo vectorial y el rasterizador, la base de los controles VCL y FMX, y los tokens de los temas claro y oscuro con los aplicadores de tema.

Stepper

Un control stepper posee un TsgcStepperModel y publica sus propiedades. La propiedad **Model** da acceso al resto: Layout, HitTest, IsStepClickable, IsConnectorDone, NextStep y PrevStep. Un cambio del modelo programa un fotograma nuevo, nunca dibuja directamente, de modo que muchos cambios seguidos cuestan un solo repintado.

Asistente y páginas

Un asistente es un contenedor de páginas, como TPageControl con sus páginas TTabSheet. Las páginas son controles hijos del asistente, en el orden de los controles, y solo la página activa es visible. Cada página contiene controles normales: campos de edición, casillas de verificación, etiquetas, marcos o cualquier otro control.
El asistente posee dos controles hijos internos que no se guardan en el flujo y no se pueden seleccionar en el diseñador: la **banda del stepper**, un stepper cuyos pasos reflejan las páginas (Caption, Description, StepGlyph, y Skip como estado omitido), y la **barra de botones** con Back, Next (Finish en la última página) y Cancel. El propio asistente establece los estados de la banda, por lo que AutoStates de la banda es False. Ambos están disponibles en tiempo de ejecución a través de las propiedades **Stepper** y **ButtonBar**.

Dibujo

Los pasos, los conectores y los botones los dibuja el renderizador. Todos los tamaños son píxeles lógicos multiplicados por la escala del monitor, de modo que los controles se ven nítidos con cualquier DPI.

Consulte también

- Plataformas
- TsgcStepperModel
- Temas

Ediciones

sgcWizard se compila desde el mismo código fuente para todas las ediciones, y todas las ediciones tienen todas las características.

Edición	Incluye
Registered	Todo: los asistentes, las páginas y los steppers VCL y FMX, la validación, las páginas omitidas, los estados de los pasos, los temas, la compatibilidad con el tiempo de diseño y la salida PNG y SVG sin interfaz.
Trial	Las mismas características que Registered. Una compilación con SGC_TRIAL muestra un mensaje de que es una versión de prueba cuando se crea un componente después de que haya pasado la ventana de publicación indicada en la compilación. Nada más cambia.
Community	Las mismas características que Registered, sin aviso de prueba.

La edición es una decisión en tiempo de compilación en sgcVer.inc. SGC_COMMUNITY también desactiva SGC_TRIAL. No es un interruptor en tiempo de ejecución, y el código escrito para una edición se compila con las demás.

Consulte también

- Plataformas
- Introducción

Instalación

sgcWizard forma parte de la distribución sgcUI. Un único instalador cubre los productos de sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard) y solo instala los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y de C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcWizard** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que usted posee y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos y después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcWizard

sgcUICore, la base compartida de los productos de disposición y de sgcCharts, se incluye con sgcWizard. No tiene casilla propia. El instalador la instala automáticamente siempre que se selecciona sgcWizard. sgcWizard no tiene paquete de puente de sgcWebSockets.

Carpetas instaladas

Todos los productos se instalan en una única carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado. Es la única ruta de bibliotecas de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Wizard`	Las demos de sgcWizard. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de bibliotecas y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore se encuentra en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala todos los productos en modo de prueba, incluido sgcWizard.
- **Community**: el instalador gratuito Community de sgcUI incluye sgcWizard. Sus productos son sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar y sgcWizard, mientras que sgcPDF y sgcMaps no forman parte de Community. La compilación Community de sgcWizard tiene las mismas características que la Registered. Consulte Ediciones.

Instalaciones independientes anteriores

Si el instalador encuentra una instalación independiente anterior de sgcUICore, como la que escribió el instalador independiente de sgcCharts, muestra las rutas de bibliotecas y los paquetes de diseño de esa instalación en la página **Previous installations**. Al continuar, los elimina del IDE antes de compilar los nuevos paquetes, de modo que nunca haya dos copias de una unidad en la ruta.

Compilar los paquetes manualmente

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de versión de Delphi, por ejemplo `sgcWizardD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcWizard<V>` / `dclsgcWizard<V>`	Los componentes de sgcWizard (tiempo de ejecución / tiempo de diseño).

Compílelos en orden de dependencia: `sgcUICore`, después `sgcWizard`, después `dclsgcWizard`. Después instale el paquete de diseño `dclsgcWizard` y añada `sgcUI\Source` a la ruta de bibliotecas de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` correspondiente a su versión, por ejemplo `sgcWizardC13.cbproj`, y compílelo. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

.NET

No cambia nada para .NET. sgcWizard se sigue distribuyendo como sus propios paquetes NuGet (`esegece.sgcWizard` y sus paquetes complementarios). Consulte la instalación de .NET.

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, de XE a XE8, de 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, de XE a XE8, de 10 a 10.4, 11, 12 y 13. Los paquetes generan los encabezados .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	TsgcVCLWizard, TsgcVCLWizardPage y TsgcVCLStepper en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	TsgcFMXWizard, TsgcFMXWizardPage y TsgcFMXStepper a partir de Delphi XE6 (SGC_WIZARD_FMX), en Windows, macOS, iOS y Android, y en Linux con FMX para Linux (FMXLinux). Los hints de los controles FMX necesitan 10.1 Berlin o posterior.
Linux64	Las unidades independientes del framework, el modelo y el renderizador con la salida PNG y SVG sin interfaz, a partir de Delphi 10.2 (SGC_WIZARD_LINUX).
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, y la salida PNG y SVG sin interfaz, en .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Paquetes

sgcWizard incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcWizardD13.dpk para Delphi 13 y sgcWizardC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcWizard	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, sgcUICore
sgcWizardVCL	Runtime, VCL	sgcWizard_VCL (TsgcVCLWizard, TsgcVCLWizardPage, TsgcVCLStepper)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcWizard
sgcWizardFMX	Runtime, FireMonkey, desde XE6	sgcWizard_FMX (TsgcFMXWizard, TsgcFMXWizardPage, TsgcFMXStepper)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcWizard
dclsgcWizard	Tiempo de diseño, VCL	Registra TsgcVCLWizard y TsgcVCLStepper en la página de paleta **SGC Wizard**, y el editor de componentes que añade páginas.	designide, rtl, vcl, sgcWizard, sgcWizardVCL
dclsgcWizardFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXWizard y TsgcFMXStepper en la página de paleta **SGC Wizard**, y el editor de componentes que añade páginas.	designide, rtl, fmx, sgcWizard, sgcWizardFMX, dclsgcWizard

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcWizardFMX requiere dclsgcWizard, así que instale primero dclsgcWizard.

Versiones del IDE y plataformas de destino

Delphi	sgcWizard	sgcWizardVCL	sgcWizardFMX	dclsgcWizard, dclsgcWizardFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcWizard, sgcWizardVCL	sgcWizardFMX	dclsgcWizard, dclsgcWizardFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcWizard)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcWizard)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcWizardC13.cbproj, sgcWizardVCLC13.cbproj, sgcWizardFMXC13.cbproj, dclsgcWizardC13.cbproj y dclsgcWizardFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXWizard, TsgcFMXWizardPage y TsgcFMXStepper) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcWizardVCL solo funcionan en Windows.

- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcWizard_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcWizard se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa el modelo y el renderizador con la salida PNG y SVG sin interfaz. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.
Las páginas no están en la paleta, se agregan con el verbo **Add Page**, consulte Tiempo de diseño.

Consulte también

- Ediciones
- Arquitectura
- Solución de problemas

Inicio rápido VCL

En tiempo de diseño

- Coloque un **TsgcVCLWizard** de la página **SGC Wizard** de la paleta en un formulario y asigne alClient a **Align**.
- Haga clic con el botón derecho en el asistente y elija **Add Page** una vez por cada paso. Establezca **Caption** y **Description** de cada página: son el título y la descripción de su paso.
- Coloque en cada página sus controles. Elija **Next Page** y **Previous Page** en el menú contextual, o haga clic en un paso de la banda del stepper, para mostrar otra página en el diseñador.
- Escriba un controlador de **OnBeforePageChange** que valide la página y asigne False a aAllow cuando la página no esté completa, y un controlador de **OnFinish** que use los datos.
- Ejecute la aplicación: Next y Enter avanzan, Back retrocede, Finish aparece en la última página, y Cancel o Escape disparan OnCancel.

En código

El mismo asistente creado en tiempo de ejecución. El programa se compila tal cual con `sgcUI\Source` en la ruta de búsqueda.

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TMainForm.CreateWizard;
var
  oPage: TsgcVCLWizardPage;
  oLabel: TLabel;
begin
  // the wizard fills the form: stepper band on top, button bar at the bottom
  FWizard := TsgcVCLWizard.Create(Self);
  FWizard.Parent := Self;
  FWizard.Align := alClient;

  // every page is a step of the stepper band
  oPage := FWizard.AddPage('Welcome');
  oPage.Description := 'Start here';
  oLabel := TLabel.Create(Self);
  oLabel.Parent := oPage;
  oLabel.SetBounds(24, 24, 320, 20);
  oLabel.Caption := 'This wizard creates your account.';

  oPage := FWizard.AddPage('Account');
  oPage.Description := 'Your name';
  FName := TEdit.Create(Self);
  FName.Parent := oPage;
  FName.SetBounds(24, 24, 240, 24);

  oPage := FWizard.AddPage('Done');
  oPage.Description := 'Summary';

  FWizard.ActivePageIndex := 0;
  FWizard.OnBeforePageChange := DoBeforePageChange;
  FWizard.OnFinish := DoFinish;
  FWizard.OnCancel := DoCancel;
end;

procedure TMainForm.DoBeforePageChange(Sender: TObject;
  aFromIndex, aToIndex: Integer; var aAllow: Boolean);
begin
  // leaving the Account page forwards needs a name
```

```
  if (aFromIndex = 1) and (aToIndex > aFromIndex) then
    aAllow := Trim(FName.Text) <> '';
end;

procedure TMainForm.DoFinish(Sender: TObject);
begin
  ShowMessage('Welcome, ' + FName.Text);
  Close;
end;

procedure TMainForm.DoCancel(Sender: TObject);
begin
  Close;
end;
```

Consulte también

- TsgcVCLWizard
- TsgcVCLWizardPage
- Validación
- Navegación y páginas omitidas

Inicio rápido FMX

Los controles de FireMonkey necesitan Delphi XE6 o posterior. Se usan exactamente igual que los de VCL: coloque un **TsgcFMXWizard**, agregue las páginas con el verbo **Add Page**, coloque en cada página sus controles y gestione **OnBeforePageChange** y **OnFinish**. Las páginas se alinean a Client dentro del asistente. Un toque en un botón o en un paso completado funciona como un clic.

En código

```
uses
  System.SysUtils, System.Classes, FMX.Forms, FMX.Types, FMX.Controls,
  FMX.StdCtrls, FMX.Edit, FMX.Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_FMX;

procedure TMainForm.CreateWizard;
var
  oPage: TsgcFMXWizardPage;
  oLabel: TLabel;
begin
  Wizard := TsgcFMXWizard.Create(Self);
  Wizard.Parent := Self;
  Wizard.Align := TAlignLayout.Client;

  oPage := Wizard.AddPage('Welcome');
  oPage.Description := 'Start here';
  oLabel := TLabel.Create(Self);
  oLabel.Parent := oPage;
  oLabel.SetBounds(24, 24, 320, 20);
  oLabel.Text := 'This wizard creates your account.';

  oPage := Wizard.AddPage('Account');
  oPage.Description := 'Your name';
  txtName := TEdit.Create(Self);
  txtName.Parent := oPage;
  txtName.SetBounds(24, 24, 240, 24);

  oPage := Wizard.AddPage('Done');
  oPage.Description := 'Summary';

  Wizard.ActivePageIndex := 0;
  Wizard.OnBeforePageChange := WizardBeforePageChange;
  Wizard.OnFinish := WizardFinish;
end;

procedure TMainForm.WizardBeforePageChange(Sender: TObject;
  aFromIndex, aToIndex: Integer; var aAllow: Boolean);
begin
  if (aFromIndex = 1) and (aToIndex > aFromIndex) then
    aAllow := Trim(txtName.Text) <> '';
end;

procedure TMainForm.WizardFinish(Sender: TObject);
begin
  ShowMessage('Welcome, ' + txtName.Text);
  Close;
end;
```

Consulte también

- TsgcFMXWizard

- TsgcFMXWizardPage
- Plataformas

Inicio rápido C++Builder

Instale los paquetes de sgcWizard para su versión de C++Builder, agregue `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de inclusión y de bibliotecas del proyecto, y use los componentes de la página de paleta **SGC Wizard** exactamente igual que en Delphi. El instalador añade ambas rutas al IDE por usted. Los encabezados .hpp y los archivos .lib generados de sgcWizard y sgcUICore están en esa única carpeta, por lo que sgcUICore no necesita una ruta propia.

En código

OnBeforePageChange y OnFinish son eventos de tipo closure, por lo que sus controladores son funciones miembro __fastcall de una clase. El parámetro var aAllow es una referencia bool.

Consulte también

- TsgcVCLWizard
- Inicio rápido VCL
- Demos

Demos

Las demos están en la carpeta **Demos** de la instalación, con los proyectos de Delphi en Demos\Delphi y las versiones para C++Builder de las demos VCL en Demos\CBuilder. Cada carpeta de demo tiene un readme.html que explica qué hace la demo y cómo funciona.

Demo	Componentes	Qué muestra
Onboarding	TsgcVCLWizard, TsgcVCLWizardPage	Un asistente de alta de cuenta con cuatro páginas: Welcome, Account, Options y Summary. OnBeforePageChange comprueba el nombre y el correo electrónico de la página Account y veta Next mientras no sean válidos, una casilla de verificación omite la página Options, OnPageChanged rellena el resumen y OnFinish lo muestra. Un interruptor de tema claro y oscuro y un combo de StepperPosition (arriba, izquierda o ninguna). Onboarding.exe /test result.txt controla el asistente y escribe una línea PASS o FAIL por comprobación. Delphi y C++Builder.
Stepper	TsgcVCLStepper	Dos steppers independientes, uno horizontal y otro vertical, con los mismos cinco pasos. Asignar ActiveIndex completa los pasos anteriores, unos botones marcan el paso activo como wssError o wssSkipped, y un clic en un paso completado devuelve ambos steppers a él mediante OnStepClick. Un interruptor de tema claro y oscuro. Delphi y C++Builder.
OnboardingFMX	TsgcFMXWizard, TsgcFMXWizardPage	El flujo Onboarding en FireMonkey: la misma validación, la misma página Options omitida, un resumen al finalizar, una casilla de verificación de tema oscuro y los mismos modificadores /test y /snapshot. Delphi.

Las demos de C++Builder incluyen los encabezados sgcWizard_*.hpp y sgcUICore_*.hpp y enlazan la biblioteca estática del paquete de ejecución de C++Builder. Se compilan con el compilador C++ clásico, Win32 y Win64.

Consulte también

- Inicio rápido VCL
- Validación
- TsgcVCLStepper

TsgcVCLWizard

Unidad: sgcWizard_VCL
TsgcVCLWizard es el asistente VCL, solo para Windows. Sus páginas son los controles hijos TsgcVCLWizardPage, la banda del stepper muestra un paso por página y la barra de botones se mueve entre ellas. Hereda de TCustomControl y publica Align, Anchors, Constraints, Enabled, Font, ParentFont, ShowHint, ParentShowHint, TabOrder, TabStop y Visible.

Propiedades

Propiedad	Tipo	Descripción
ActivePageIndex	Integer	El índice de la página visible. Asignarlo por código cambia de página directamente, sin eventos. Por defecto 0.
ActivePage	clase de la página	Pública. La página visible, o nil. Asignarla cambia de página sin eventos.
PageCount	Integer	Pública, de solo lectura. El número de páginas.
Pages[Index]	clase de la página	Pública, de solo lectura. Las páginas, en el orden de los controles.
StepperPosition	TsgcWizardStepperPosition	wspTop (por defecto) muestra una banda de stepper horizontal en la parte superior, wspLeft una banda vertical a la izquierda, wspNone la oculta. Consulte Disposición del stepper.
BackCaption, NextCaption, FinishCaption, CancelCaption	string	Los títulos de los botones. Por defecto Back, Next, Finish y Cancel. Solo se almacenan cuando cambian. Consulte Títulos de los botones.
ShowCancel	Boolean	Muestra el botón Cancel. Por defecto True.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. El asistente lo propaga a la banda, a la barra de botones y a los controles hijos de las páginas. Consulte Temas.
AccentColor	TsgcUIColor	Un acento $AARRGGBB para los pasos activos y completados, los conectores hechos y el botón por defecto. 0 (por defecto) usa el acento del tema.
Stepper	clase del stepper	Pública, de solo lectura. La banda interna del stepper.
ButtonBar	clase de la barra de botones	Pública, de solo lectura. La barra de botones interna. ButtonCount, Buttons, ButtonRect y ButtonAt describen los botones tal como se dibujan.

Métodos

Método	Devuelve	Descripción

Next	Boolean	Pasa a la siguiente página que no se omite, disparando OnBeforePageChange y OnPageChanged. En la última página llama a Finish. True cuando la página cambió o el asistente terminó.
Back	Boolean	Pasa a la página anterior que no se omite, con los mismos eventos.
GoToPage(aIndex)	Boolean	Pasa a cualquier página que no se omite, con los mismos eventos. False para una página omitida o un índice fuera de rango.
Finish	Boolean	Dispara OnBeforePageChange con aToIndex = PageCount y, si se permite, OnFinish.
Cancel		Dispara OnCancel.
NextPageIndex(aIndex), PrevPageIndex(aIndex)	Integer	La página siguiente o anterior a aIndex que no se omite, -1 si no hay ninguna.
IsLastPage	Boolean	True cuando no se puede llegar a ninguna página posterior a la activa, de modo que Next muestra Finish.
IndexOfPage(aPage)	Integer	El índice de una página, -1 cuando no es una página de este asistente.
AddPage(aCaption, aOwner = nil)	clase de la página	Crea una página con ese Caption, propiedad de aOwner (nil usa el propietario del asistente), y la coloca en último lugar.

Eventos

Evento	Tipo	Descripción
OnBeforePageChange	TsgcWizardPageChangeEvent	Antes de cada cambio por Next, Back, GoToPage, Finish o un clic en un paso. aFromIndex, aToIndex y var aAllow: asigne False a aAllow para permanecer en la página. aToIndex = PageCount significa Finish. Consulte Validación.
OnPageChanged	TNotifyEvent	Después de que la página cambió.
OnFinish	TNotifyEvent	Después de que se permitió Finish. Consulte Finalizar y cancelar.
OnCancel	TNotifyEvent	Cancel, el botón Cancel o Escape.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.BuildPages;
var
  oPage: TsgcVCLWizardPage;
begin
  Wizard := TsgcVCLWizard.Create(Self);
  Wizard.Parent := Self;
  Wizard.Align := alClient;

  oPage := Wizard.AddPage('Welcome');
```

```
  chkSkipOptions := TCheckBox.Create(Self);
  chkSkipOptions.Parent := oPage;
  chkSkipOptions.SetBounds(24, 24, 240, 20);
  chkSkipOptions.Caption := 'Use the default options';
  chkSkipOptions.OnClick := SkipClick;

  oPage := Wizard.AddPage('Account');
  txtEmail := TEdit.Create(Self);
  txtEmail.Parent := oPage;
  txtEmail.SetBounds(24, 24, 240, 24);

  Wizard.AddPage('Options');
  Wizard.AddPage('Summary');

  Wizard.StepperPosition := wspTop;
  Wizard.OnBeforePageChange := WizardBeforePageChange;
  Wizard.OnFinish := WizardFinish;
  Wizard.OnCancel := WizardCancel;
  Wizard.ActivePageIndex := 0;
end;
```

Consulte también

- TsgcVCLWizardPage
- TsgcFMXWizard
- Navegación y páginas omitidas
- Teclado

TsgcVCLWizardPage

Unidad: sgcWizard_VCL
TsgcVCLWizardPage es una página de un TsgcVCLWizard. Hereda de TCustomControl, acepta controles hijos y también publica Enabled, Font, ParentFont, Hint, ShowHint, ParentShowHint, PopupMenu, OnEnter, OnExit y OnResize. Una página solo se puede colocar en un asistente: asignar su Parent a cualquier otro control genera EsgcWizardError.

Propiedades

Propiedad	Tipo	Descripción
Caption	string	El título del paso de la página.
Description	string	La descripción del paso, dibujada debajo del título mientras ShowDescriptions de la banda es True, y el hint del paso.
Skip	Boolean	Next y Back saltan la página y la banda dibuja su paso como wssSkipped. Por defecto False. Consulte Navegación y páginas omitidas.
StepGlyph	string	Un texto corto dibujado en el círculo del paso. Vacío (por defecto) dibuja el número del paso.
Visible	Boolean	La gestiona el asistente: solo la página activa es visible. No se almacena.
Wizard	clase del asistente	Pública, de solo lectura. El asistente de la página, nil cuando no tiene ninguno.
PageIndex	Integer	Pública, de solo lectura. La posición entre las páginas del asistente, -1 cuando no está en un asistente.

El asistente pinta la página con el color Surface de su tema y aplica el tema a los controles hijos directos de la página, consulte Temas.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.PageExample;
var
  oPage: TsgcVCLWizardPage;
begin
  oPage := TsgcVCLWizardPage.Create(Self);
  oPage.Caption := 'Licence';
  oPage.Description := 'Accept the terms';
  oPage.StepGlyph := 'L';
  // a page can only be placed on a wizard, it becomes the last page
  oPage.Parent := Wizard;
  Caption := Format('Page %d of %d', [oPage.PageIndex + 1,
    Wizard.PageCount]);
end;
```

Consulte también

- TsgcVCLWizard
- Tiempo de diseño
- Solución de problemas

TsgcFMXWizard

Unidad: sgcWizard_FMX
TsgcFMXWizard es el asistente de FireMonkey, a partir de Delphi XE6. Tiene las mismas propiedades, métodos y eventos que TsgcVCLWizard. Las páginas son los hijos TsgcFMXWizardPage, alineados a Client, la banda del stepper se alinea a MostTop o MostLeft y la barra de botones a MostBottom. Las posiciones y los tamaños son unidades lógicas de FMX. Publica Align, Anchors, CanFocus, ClipChildren, Enabled, Height, HitTest, Margins, Opacity, Padding, PopupMenu, Position, Size, TabOrder, TabStop, Visible y Width, ShowHint y ParentShowHint a partir de 10.1 Berlin, y OnEnter, OnExit, OnKeyDown, OnKeyUp y OnResize.

Propiedades

Propiedad	Tipo	Descripción
ActivePageIndex	Integer	El índice de la página visible. Asignarlo por código cambia de página directamente, sin eventos. Por defecto 0.
ActivePage	clase de la página	Pública. La página visible, o nil. Asignarla cambia de página sin eventos.
PageCount	Integer	Pública, de solo lectura. El número de páginas.
Pages[Index]	clase de la página	Pública, de solo lectura. Las páginas, en el orden de los controles.
StepperPosition	TsgcWizardStepperPosition	wspTop (por defecto) muestra una banda de stepper horizontal en la parte superior, wspLeft una banda vertical a la izquierda, wspNone la oculta. Consulte Disposición del stepper.
BackCaption, NextCaption, FinishCaption, CancelCaption	string	Los títulos de los botones. Por defecto Back, Next, Finish y Cancel. Solo se almacenan cuando cambian. Consulte Títulos de los botones.
ShowCancel	Boolean	Muestra el botón Cancel. Por defecto True.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. El asistente lo propaga a la banda, a la barra de botones y a los controles hijos de las páginas. Consulte Temas.
AccentColor	TsgcUIColor	Un acento $AARRGGBB para los pasos activos y completados, los conectores hechos y el botón por defecto. 0 (por defecto) usa el acento del tema.
Stepper	clase del stepper	Pública, de solo lectura. La banda interna del stepper.
ButtonBar	clase de la barra de botones	Pública, de solo lectura. La barra de botones interna. ButtonCount, Buttons, ButtonRect y ButtonAt describen los botones tal como se dibujan.

Métodos

Método	Devuelve	Descripción

Next	Boolean	Pasa a la siguiente página que no se omite, disparando OnBeforePageChange y OnPageChanged. En la última página llama a Finish. True cuando la página cambió o el asistente terminó.
Back	Boolean	Pasa a la página anterior que no se omite, con los mismos eventos.
GoToPage(aIndex)	Boolean	Pasa a cualquier página que no se omite, con los mismos eventos. False para una página omitida o un índice fuera de rango.
Finish	Boolean	Dispara OnBeforePageChange con aToIndex = PageCount y, si se permite, OnFinish.
Cancel		Dispara OnCancel.
NextPageIndex(aIndex), PrevPageIndex(aIndex)	Integer	La página siguiente o anterior a aIndex que no se omite, -1 si no hay ninguna.
IsLastPage	Boolean	True cuando no se puede llegar a ninguna página posterior a la activa, de modo que Next muestra Finish.
IndexOfPage(aPage)	Integer	El índice de una página, -1 cuando no es una página de este asistente.
AddPage(aCaption, aOwner = nil)	clase de la página	Crea una página con ese Caption, propiedad de aOwner (nil usa el propietario del asistente), y la coloca en último lugar.

Eventos

Evento	Tipo	Descripción
OnBeforePageChange	TsgcWizardPageChangeEvent	Antes de cada cambio por Next, Back, GoToPage, Finish o un clic en un paso. aFromIndex, aToIndex y var aAllow: asigne False a aAllow para permanecer en la página. aToIndex = PageCount significa Finish. Consulte Validación.
OnPageChanged	TNotifyEvent	Después de que la página cambió.
OnFinish	TNotifyEvent	Después de que se permitió Finish. Consulte Finalizar y cancelar.
OnCancel	TNotifyEvent	Cancel, el botón Cancel o Escape.

Ejemplo

```
uses
  System.SysUtils, System.Classes, FMX.Forms, FMX.Types, FMX.Controls,
  FMX.StdCtrls, FMX.Edit, FMX.Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_FMX;

procedure TMainForm.WizardExample;
begin
  // the stepper band on the left, a dark theme and localised captions
  Wizard.StepperPosition := wspLeft;
  Wizard.Theme := utDark;
  Wizard.NextCaption := 'Continue';
  Wizard.ShowCancel := False;
  if Wizard.Next then
```

```
    Caption := Wizard.ActivePage.Caption;
end;
```

Consulte también

- TsgcFMXWizardPage
- Inicio rápido FMX
- Temas

TsgcFMXWizardPage

Unidad: sgcWizard_FMX
TsgcFMXWizardPage es una página de un TsgcFMXWizard, alineada a Client dentro de él. También publica ClipChildren, Enabled, HitTest, Padding, PopupMenu, OnEnter, OnExit y OnResize, y ShowHint y ParentShowHint a partir de 10.1 Berlin. Una página solo se puede colocar en un asistente: asignar su Parent a cualquier otro objeto genera EsgcWizardError.

Propiedades

Propiedad	Tipo	Descripción
Caption	string	El título del paso de la página.
Description	string	La descripción del paso, dibujada debajo del título mientras ShowDescriptions de la banda es True, y el hint del paso.
Skip	Boolean	Next y Back saltan la página y la banda dibuja su paso como wssSkipped. Por defecto False. Consulte Navegación y páginas omitidas.
StepGlyph	string	Un texto corto dibujado en el círculo del paso. Vacío (por defecto) dibuja el número del paso.
Visible	Boolean	La gestiona el asistente: solo la página activa es visible. No se almacena.
Wizard	clase del asistente	Pública, de solo lectura. El asistente de la página, nil cuando no tiene ninguno.
PageIndex	Integer	Pública, de solo lectura. La posición entre las páginas del asistente, -1 cuando no está en un asistente.

Ejemplo

```
uses
  System.SysUtils, System.Classes, FMX.Forms, FMX.Types, FMX.Controls,
  FMX.StdCtrls, FMX.Edit, FMX.Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_FMX;

procedure TMainForm.PageExample;
var
  oPage: TsgcFMXWizardPage;
begin
  oPage := TsgcFMXWizardPage.Create(Self);
  oPage.Caption := 'Options';
  oPage.Description := 'Optional settings';
  oPage.Skip := True;
  // a page can only be placed on a wizard; it is aligned Client
  oPage.Parent := Wizard;
  Caption := Format('Page %d of %d', [oPage.PageIndex + 1,
    Wizard.PageCount]);
end;
```

Consulte también

- TsgcFMXWizard

- TsgcVCLWizardPage
- Temas

TsgcVCLStepper

Unidad: sgcWizard_VCL
TsgcVCLStepper es el indicador de pasos VCL, utilizable por sí solo. Muestra sus pasos como círculos numerados unidos por conectores, en horizontal o en vertical. El hint de un paso es su Description. Mientras el control tiene el foco, Izquierda y Derecha (Arriba y Abajo en vertical) pasan al paso visible y habilitado anterior o siguiente a través del mismo veto de OnStepClick, mientras Clickable sea True. También publica Align, Anchors, Constraints, Enabled, ShowHint, ParentShowHint, TabOrder, TabStop y Visible.

Propiedades

Propiedad	Tipo	Descripción
Steps	TsgcWizardSteps	Los pasos.
Orientation	TsgcWizardOrientation	woHorizontal (por defecto) o woVertical.
ActiveIndex	Integer	El paso activo. Por defecto 0. Asignarlo por código no dispara OnActiveChange.
AutoStates	Boolean	Por defecto True: al asignar ActiveIndex se completan los pasos visibles anteriores, salvo que sean wssError o wssSkipped, y los posteriores pasan a wssPending. Consulte Estados de los pasos.
ShowDescriptions	Boolean	Dibuja la Description de cada paso debajo de su título. Por defecto True.
CircleSize	Integer	El diámetro de los círculos de los pasos, en píxeles lógicos. Por defecto 28.
ConnectorThickness	Integer	El grosor de los conectores, en píxeles lógicos. Por defecto 2.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom.
AccentColor	TsgcUIColor	Un acento $AARRGGBB, 0 (por defecto) usa el acento del tema.
Clickable	Boolean	Un clic, o las teclas de flecha, pueden activar un paso completado, o un paso con error anterior al activo. Por defecto True.
Model	TsgcStepperModel	Pública, de solo lectura. El modelo del control.
Wizard	clase del asistente	Pública, de solo lectura. El asistente que aloja el stepper como su banda, nil para un stepper independiente.

Métodos

Método	Devuelve	Descripción
StepAt(X, Y)	Integer	El paso situado bajo un punto del cliente, -1 si no hay ninguno.

StepRect(aIndex)	TRect (TRectF en FMX)	El círculo y la etiqueta de un paso en coordenadas del cliente, vacío cuando el paso no se muestra.

Eventos

Evento	Tipo	Descripción
OnStep-Click	TsgcWizardS-tepClickEvent	Se hizo clic en un paso con clic habilitado, o una tecla de flecha se mueve a él. aIndex y var aAllow: asigne False a aAllow para mantener el paso activo.
OnActive-Change	TNotifyEvent	El paso activo cambió por un clic o una tecla. Nunca se dispara cuando ActiveIndex se asigna por código.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.StepperExample;
begin
  Stepper := TsgcVCLStepper.Create(Self);
  Stepper.Parent := Self;
  Stepper.Align := alTop;
  Stepper.Height := 80;
  Stepper.Steps.AddStep('Cart', 'Review the items');
  Stepper.Steps.AddStep('Shipping', 'Address and carrier');
  Stepper.Steps.AddStep('Payment', 'Card or transfer');
  Stepper.Steps.AddStep('Done', '', 'OK');
  // AutoStates: the steps before ActiveIndex become completed
  Stepper.ActiveIndex := 1;
  Stepper.OnStepClick := StepperStepClick;
end;

procedure TFeaturesForm.StepperStepClick(Sender: TObject; aIndex: Integer;
  var aAllow: Boolean);
begin
  // aAllow False keeps the current step active
  aAllow := aIndex > 0;
end;
```

Consulte también

- TsgcFMXStepper
- Estados de los pasos
- Disposición del stepper

TsgcFMXStepper

Unidad: sgcWizard_FMX
TsgcFMXStepper es el indicador de pasos de FireMonkey, a partir de Delphi XE6, con las mismas propiedades, métodos y eventos que TsgcVCLStepper. A partir de 10.1 Berlin el Hint del control sigue a la Description del paso situado bajo el ratón. También publica Align, Anchors, CanFocus, Enabled, Height, HitTest, Margins, Opacity, Padding, PopupMenu, Position, Size, TabOrder, TabStop, Visible y Width, y OnEnter, OnExit, OnKeyDown, OnKeyUp, OnMouseDown, OnMouseMove, OnMouseUp y OnResize.

Propiedades

Propiedad	Tipo	Descripción
Steps	TsgcWizardSteps	Los pasos.
Orientation	TsgcWizardOrientation	woHorizontal (por defecto) o woVertical.
ActiveIndex	Integer	El paso activo. Por defecto 0. Asignarlo por código no dispara OnActiveChange.
AutoStates	Boolean	Por defecto True: al asignar ActiveIndex se completan los pasos visibles anteriores, salvo que sean wssError o wssSkipped, y los posteriores pasan a wssPending. Consulte Estados de los pasos.
ShowDescriptions	Boolean	Dibuja la Description de cada paso debajo de su título. Por defecto True.
CircleSize	Integer	El diámetro de los círculos de los pasos, en píxeles lógicos. Por defecto 28.
ConnectorThickness	Integer	El grosor de los conectores, en píxeles lógicos. Por defecto 2.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom.
AccentColor	TsgcUIColor	Un acento $AARRGGBB, 0 (por defecto) usa el acento del tema.
Clickable	Boolean	Un clic, o las teclas de flecha, pueden activar un paso completado, o un paso con error anterior al activo. Por defecto True.
Model	TsgcStepperModel	Pública, de solo lectura. El modelo del control.
Wizard	clase del asistente	Pública, de solo lectura. El asistente que aloja el stepper como su banda, nil para un stepper independiente.

Métodos

Método	Devuelve	Descripción
StepAt(X, Y)	Integer	El paso situado bajo un punto del cliente, -1 si no hay ninguno.

StepRect(aIndex)	TRect (TRectF en FMX)	El círculo y la etiqueta de un paso en coordenadas del cliente, vacío cuando el paso no se muestra.

Eventos

Evento	Tipo	Descripción
OnStep-Click	TsgcWizardS-tepClickEvent	Se hizo clic en un paso con clic habilitado, o una tecla de flecha se mueve a él. aIndex y var aAllow: asigne False a aAllow para mantener el paso activo.
OnActive-Change	TNotifyEvent	El paso activo cambió por un clic o una tecla. Nunca se dispara cuando ActiveIndex se asigna por código.

Ejemplo

```
uses
  System.SysUtils, System.Classes, FMX.Forms, FMX.Types, FMX.Controls,
  FMX.StdCtrls, FMX.Edit, FMX.Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_FMX;

procedure TMainForm.StepperExample;
begin
  Stepper := TsgcFMXStepper.Create(Self);
  Stepper.Parent := Self;
  Stepper.Align := TAlignLayout.Top;
  Stepper.Height := 80;
  Stepper.Steps.AddStep('Cart', 'Review the items');
  Stepper.Steps.AddStep('Shipping', 'Address and carrier');
  Stepper.Steps.AddStep('Payment', 'Card or transfer');
  Stepper.ActiveIndex := 2;
  Stepper.Steps[1].State := wssError;
end;
```

Consulte también

- TsgcVCLStepper
- Estados de los pasos
- Plataformas

TsgcWizardStep

Unidad: sgcWizard_Model
TsgcWizardStep es un paso de un stepper, un elemento de TsgcWizardSteps. En un asistente los pasos de la banda se crean a partir de las páginas, por lo que la aplicación establece en su lugar las propiedades de las páginas.

Propiedades

Propiedad	Tipo	Descripción
Caption	string	El título dibujado debajo del círculo (horizontal) o a su derecha (vertical).
Description	string	La segunda línea de la etiqueta, dibujada mientras ShowDescriptions es True, y el hint del paso.
Glyph	string	Un texto corto dibujado en el círculo. Vacío dibuja el número del paso.
State	TsgcWizardStepState	wssPending (por defecto), wssActive, wssCompleted, wssError o wssSkipped. Consulte Estados de los pasos.
Enabled	Boolean	Un paso deshabilitado se dibuja con colores atenuados y no admite clic. Por defecto True.
Visible	Boolean	Un paso oculto no recibe hueco ni número, y conserva su estado. Por defecto True.
Tag	NativeInt (Integer antes de XE2)	De uso libre para la aplicación. Por defecto 0.
Shown, Number, CircleRect, LabelRect	Boolean, Integer, TsgcUIRect	Pública, de solo lectura. Los resultados del último Layout: si el paso obtuvo un hueco, su número (base 1) entre los pasos visibles y los rectángulos de su círculo y su etiqueta en píxeles de dispositivo.

Métodos

Método	Devuelve	Descripción
CircleText	string	El texto dibujado en el círculo: Glyph, o Number cuando Glyph está vacío.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.StepsExample;
var
  oStep: TsgcWizardStep;
begin
  Stepper.Steps.Clear;
  oStep := Stepper.Steps.Add;
```

```
  oStep.Caption := 'Upload';
  oStep.Description := 'Choose the files';
  oStep := Stepper.Steps.AddStep('Review', 'Check the preview');
  oStep.Glyph := 'R';
  // a hidden step gets no slot and no number
  Stepper.Steps.AddStep('Advanced').Visible := False;
  Stepper.Steps.AddStep('Publish');
  Stepper.ActiveIndex := 0;
end;
```

Consulte también

- TsgcWizardSteps
- TsgcStepperModel

TsgcWizardSteps

Unidad: sgcWizard_Model
TsgcWizardSteps es la colección propia de elementos TsgcWizardStep que hay tras la propiedad Steps de un stepper y de TsgcStepperModel. Se edita en tiempo de diseño con el editor de colecciones estándar.

Métodos

Método	Devuelve	Descripción
Add	TsgcWizardStep	Agrega un paso vacío.
AddStep(aCaption, aDescription = '', aGlyph = '')	TsgcWizardStep	Agrega un paso con su título, su descripción y su glifo y lo devuelve.
Items[Index]	TsgcWizardStep	La propiedad indexada por defecto.

Agregar o quitar un paso recalcula los estados automáticos mientras AutoStates es True.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.StepperExample;
begin
  Stepper := TsgcVCLStepper.Create(Self);
  Stepper.Parent := Self;
  Stepper.Align := alTop;
  Stepper.Height := 80;
  Stepper.Steps.AddStep('Cart', 'Review the items');
  Stepper.Steps.AddStep('Shipping', 'Address and carrier');
  Stepper.Steps.AddStep('Payment', 'Card or transfer');
  Stepper.Steps.AddStep('Done', '', 'OK');
  // AutoStates: the steps before ActiveIndex become completed
  Stepper.ActiveIndex := 1;
  Stepper.OnStepClick := StepperStepClick;
end;

procedure TFeaturesForm.StepperStepClick(Sender: TObject; aIndex: Integer;
  var aAllow: Boolean);
begin
  // aAllow False keeps the current step active
  aAllow := aIndex > 0;
end;
```

Consulte también

- TsgcWizardStep
- TsgcVCLStepper

TsgcStepperModel

Unidad: sgcWizard_Model
TsgcStepperModel es el modelo de un stepper independiente del framework: los pasos con su estado, la orientación, el paso activo, el paso bajo el cursor y la disposición de los círculos, las etiquetas y los conectores, con su prueba de impacto. Los steppers VCL y FMX poseen uno y publican sus propiedades, y el renderizado sin interfaz lo dibuja sin ventana. Está pensado para usarse desde el hilo principal.

Propiedades

Propiedad	Tipo	Descripción
Steps, Orientation, ActiveIndex, AutoStates, ShowDescriptions, CircleSize, ConnectorThickness, Theme, AccentColor, Clickable		Las propiedades publicadas de los steppers, consulte TsgcVCLStepper.
HotIndex	Integer	El paso situado bajo el ratón, establecido por el control. Un paso con clic habilitado bajo el cursor recibe una ligera superposición.
ConnectorCount, Connectors[Index]	Integer, TsgcWizardConnector	Los conectores del último Layout: FromIndex, ToIndex y la línea X1, Y1, X2, Y2 en píxeles de dispositivo.
LayoutWidth, LayoutHeight, LayoutScale	Integer, Double	El tamaño y la escala del último Layout.
OnChange	TNotifyEvent	Cualquier cambio del modelo. Layout nunca lo dispara.

Métodos

Método	Devuelve	Descripción
Layout(aWidth, aHeight, aScale = 1) y sobrecargas con una función de medición o un lienzo		Calcula los rectángulos de los pasos y de los conectores para un stepper de aWidth por aHeight píxeles de dispositivo. Cada tamaño lógico se multiplica por aScale. Sin una función de medición ni un lienzo, los anchos de las etiquetas se estiman.
HitTest(aX, aY)	Integer	El paso cuyo círculo o etiqueta contiene el punto del último Layout, -1 si no hay ninguno. También se notifican los pasos deshabilitados.
IsStepClickable(aIndex)	Boolean	Con Clickable activo, un paso visible y habilitado que no es el activo admite clic cuando es wssCompleted, o wssError anterior al paso activo.
IsConnectorDone(aToIndex)	Boolean	La regla del conector, consulte Estados de los pasos.

ApplyAutoStates		Aplica los estados automáticos ahora, sea cual sea el valor de AutoStates.
VisibleCount	Integer	El número de pasos visibles.
NextStep(aIndex), PrevStep(aIndex)	Integer	El paso visible y habilitado siguiente o anterior, -1 si no hay ninguno.
GetAccent	TsgcUIColor	El color de acento en uso: AccentColor, o el acento del tema cuando es 0.

Ejemplo

El programa de consola siguiente imprime Clickable: Cart, Visible steps: 3, After Shipping: 2, Connectors: 2 e Into Shipping done: TRUE.

```
uses
  SysUtils, Classes,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_Render;

procedure ModelExample;
var
  oModel: TsgcStepperModel;
  vIndex: Integer;
begin
  oModel := TsgcStepperModel.Create;
  try
    oModel.Steps.AddStep('Cart');
    oModel.Steps.AddStep('Shipping');
    oModel.Steps.AddStep('Payment');
    oModel.ActiveIndex := 1;

    // step rectangles for a stepper of 480 x 80 pixels at scale 1
    oModel.Layout(480, 80);
    vIndex := oModel.HitTest(oModel.Steps[0].CircleRect.Left + 2,
      oModel.Steps[0].CircleRect.Top + 2);
    if (vIndex >= 0) and oModel.IsStepClickable(vIndex) then
      WriteLn('Clickable: ', oModel.Steps[vIndex].Caption);

    WriteLn('Visible steps: ', oModel.VisibleCount);
    WriteLn('After Shipping: ', oModel.NextStep(1));
    WriteLn('Connectors: ', oModel.ConnectorCount);
    WriteLn('Into Shipping done: ', oModel.IsConnectorDone(1));
  finally
    oModel.Free;
  end;
end;
```

Consulte también

- TsgcWizardStep
- Renderizado sin interfaz
- Estados de los pasos

Navegación y páginas omitidas

El usuario se mueve por un asistente con la barra de botones, el teclado y los pasos completados de la banda del stepper. La aplicación usa los mismos métodos.

Acción	Efecto
Next, el botón Next, Enter	Pasa a la siguiente página que no se omite. En la última página llama a Finish.
Back, el botón Back	Pasa a la página anterior que no se omite. El botón Back se deshabilita cuando no se puede llegar a ninguna página anterior.
GoToPage(aIndex)	Pasa a cualquier página que no se omite.
Un clic en un paso completado de la banda	Llama a GoToPage para su página.
ActivePageIndex o ActivePage asignados por código	Cambia de página directamente, sin eventos.

Todo movimiento salvo el cambio directo dispara OnBeforePageChange, que puede vetarlo, y después OnPageChanged. Cuando el foco estaba dentro del asistente, un cambio de página lo lleva al primer control de la página nueva, o al asistente.

Páginas omitidas

Una página cuya propiedad **Skip** es True permanece en el asistente, pero Next y Back la saltan y la banda del stepper dibuja su paso como omitido con un contorno discontinuo. Skip puede cambiar en cualquier momento, por ejemplo desde una casilla de verificación de una página anterior. NextPageIndex y PrevPageIndex devuelven la página a la que irían Next y Back, e IsLastPage es True cuando no se puede llegar a ninguna página posterior a la activa, de modo que el botón Next muestra el título Finish.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.NavigationExample;
begin
  // Next, Back and GoToPage fire the events and jump over skipped pages
  if not Wizard.Next then
    ShowMessage('The page did not change');
  Wizard.Back;
  Wizard.GoToPage(Wizard.PageCount - 1);

  // the page that Next goes to, or the last page
  if Wizard.IsLastPage then
    Caption := 'Last page'
  else
    Caption := 'Next: ' + Wizard.Pages
      [Wizard.NextPageIndex(Wizard.ActivePageIndex)].Caption;

  // setting ActivePageIndex by code switches the page without events
  Wizard.ActivePageIndex := 0;
end;
```

```
procedure TFeaturesForm.SkipClick(Sender: TObject);
begin
  // Next and Back jump over a skipped page, the stepper draws it as skipped
  Wizard.Pages[2].Skip := chkSkipOptions.Checked;
end;
```

Consulte también

- TsgcVCLWizard
- Validación
- Finalizar y cancelar

Validación

OnBeforePageChange se dispara antes de cada cambio de página hecho por Next, Back, GoToPage, Finish o un clic en un paso. Recibe aFromIndex, la página activa, aToIndex, la página que está a punto de mostrarse, y un var aAllow que empieza en True. Asigne False a aAllow para permanecer en la página, por ejemplo mientras un campo obligatorio está vacío.

- **aToIndex > aFromIndex** es todo movimiento hacia delante, Finish incluido, de modo que una sola comprobación valida la página sea cual sea la tecla o el botón que pulsó el usuario.
- **aToIndex = PageCount** significa que el usuario pulsó Finish: Finish dispara el evento con PageCount como destino, que nunca es un índice de página real.
- Los movimientos hacia atrás normalmente no necesitan validación. Compruebe aToIndex < aFromIndex para dejarlos pasar.
- Cuando se veta el cambio no ocurre nada más: la página, la banda y los botones se quedan como están, así que muestre el motivo y lleve el foco al campo.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.WizardBeforePageChange(Sender: TObject;
  aFromIndex, aToIndex: Integer; var aAllow: Boolean);
begin
  // aToIndex > aFromIndex is every forward move, Finish included
  if (aFromIndex = 1) and (aToIndex > aFromIndex) and
    (Pos('@', txtEmail.Text) = 0) then
  begin
    aAllow := False;
    ShowMessage('Enter a valid email address.');
    txtEmail.SetFocus;
  end;
  // aToIndex = PageCount means that the user pressed Finish
  if aAllow and (aToIndex = Wizard.PageCount) then
    aAllow := MessageDlg('Create the account?', mtConfirmation,
      [mbYes, mbNo], 0) = mrYes;
end;
```

Consulte también

- Navegación y páginas omitidas
- Finalizar y cancelar
- Demos

Finalizar y cancelar

En la última página el botón Next muestra el **FinishCaption** y es el botón por defecto. Pulsarlo, pulsar Enter o llamar a **Finish** dispara OnBeforePageChange con aToIndex = PageCount y, si se permite, **OnFinish**. El asistente no cierra nada por sí mismo: OnFinish decide qué ocurre, por ejemplo guarda los datos y cierra el formulario, o asigna el ModalResult de un cuadro de diálogo.
El botón Cancel, la tecla Escape y una llamada a **Cancel** disparan **OnCancel**. No tiene veto y no cambia la página. Asigne False a **ShowCancel** para ocultar el botón Cancel.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.WizardFinish(Sender: TObject);
begin
  // runs after OnBeforePageChange allowed aToIndex = PageCount
  ModalResult := mrOk;
end;

procedure TFeaturesForm.WizardCancel(Sender: TObject);
begin
  // the Cancel button, the Escape key or a call to Cancel
  if MessageDlg('Discard the changes?', mtConfirmation, [mbYes, mbNo], 0)
    = mrYes then
    ModalResult := mrCancel;
end;
```

Consulte también

- Validación
- Teclado
- Títulos de los botones

Teclado

Mientras el asistente o un control situado dentro de él tiene el foco, **Enter** es Next (Finish en la última página) y **Escape** es Cancel. Prevalece el control que quiere la tecla para sí: un TMemo conserva Enter para una nueva línea, y un botón con foco, o un TButton con Default o Cancel activado, gestiona la tecla como de costumbre.

Framework	Cómo llegan las teclas
VCL	Mediante CM_DIALOGKEY desde el formulario padre, o mediante KeyDown cuando el propio asistente tiene el foco.
FireMonkey	Mediante la cadena de teclas de diálogo del formulario (DialogKey y AfterDialogKey), o mediante KeyDown cuando el propio asistente tiene el foco.

Después de un cambio de página el foco pasa al primer control de la página nueva, o al asistente, cuando antes estaba dentro del asistente.

Teclas del stepper

Un stepper con foco pasa con Izquierda y Derecha, o con Arriba y Abajo cuando es vertical, al paso visible y habilitado anterior o siguiente. El movimiento pasa por el mismo veto de OnStepClick que un clic y solo ocurre mientras Clickable es True.

Consulte también

- Navegación y páginas omitidas
- Finalizar y cancelar
- TsgcVCLStepper

Estados de los pasos

Cada paso tiene un **State** que decide cómo se dibuja.

Estado	Se dibuja como
wssPending	Un contorno con el color del borde y el número o el glifo del paso con el color de texto atenuado.
wssActive	Un disco con el color de acento, con un suave halo de acento y el texto del paso en un color de contraste. El título va en negrita.
wssCompleted	Un disco con el color de acento y una marca de verificación.
wssError	Un disco rojo con un signo de exclamación y un título rojo.
wssSkipped	Un contorno discontinuo con el texto del paso en el color de texto atenuado.

Un paso deshabilitado se dibuja con colores atenuados.

Estados automáticos

Mientras **AutoStates** es True (por defecto), cada vez que se asigna ActiveIndex, se agrega o se quita un paso, o se activa AutoStates, los pasos visibles anteriores al activo pasan a wssCompleted salvo que sean wssError o wssSkipped, el paso activo pasa a wssActive y los pasos visibles posteriores pasan a wssPending. Los pasos ocultos conservan su estado. Con AutoStates en False, todos los estados quedan a cargo de la aplicación. En un asistente la banda tiene AutoStates en False, porque el asistente establece los estados a partir de las páginas y de su propiedad Skip.

La regla del conector

El conector entre dos pasos visibles consecutivos está hecho, y se dibuja con el color de acento, cuando el paso al que conduce es wssCompleted o wssActive, o cuando ese paso es wssSkipped o wssError y ya se ha superado, es decir, algún paso visible posterior es wssActive o wssCompleted. En caso contrario se dibuja con el color del borde. **Model.IsConnectorDone(aToIndex)** responde a la regla para el conector que conduce al paso aToIndex, de modo que una página omitida en medio de un tramo terminado del flujo no interrumpe la línea de acento.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.StatesExample;
var
  i: Integer;
begin
  // an error or skipped state survives the automatic states
  Stepper.Steps[1].State := wssError;
  Stepper.ActiveIndex := 2;

  // with AutoStates False the application sets every state
  Stepper.AutoStates := False;
  for i := 0 to Stepper.Steps.Count - 1 do
    Stepper.Steps[i].State := wssPending;
```

```
  Stepper.Steps[0].State := wssCompleted;
  Stepper.Steps[1].State := wssSkipped;
  Stepper.Steps[2].State := wssActive;

  // the connector into step 1 is done: step 1 was skipped and passed
  if Stepper.Model.IsConnectorDone(1) then
    Caption := 'Step 2 was passed';
end;
```

Consulte también

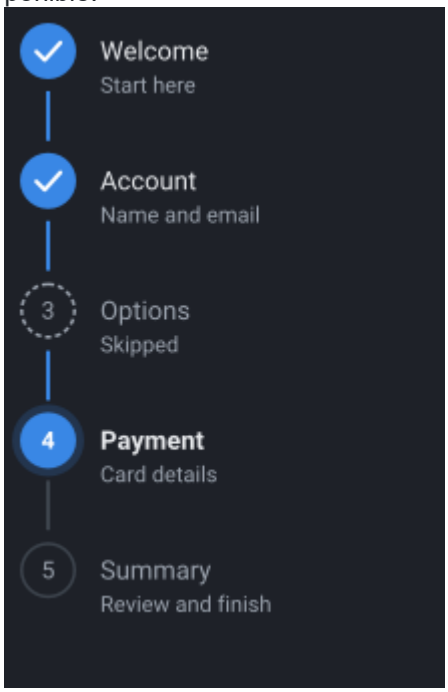
- TsgcWizardStep
- TsgcStepperModel
- Disposición del stepper

Disposición del stepper

Un stepper es horizontal o vertical, según **Orientation**. Solo los pasos visibles reciben un hueco, y su número es su posición (base 1) entre los pasos visibles.

- **woHorizontal**: el ancho se reparte en huecos iguales, el círculo se centra en su hueco, y el título y la descripción se centran debajo. El bloque se centra verticalmente.
- **woVertical**: el alto se reparte en filas iguales, el círculo se sitúa arriba a la izquierda de su fila, y la etiqueta queda a su derecha con el título centrado respecto al círculo. Los conectores bajan entre los círculos.

CircleSize (por defecto 28) y **ConnectorThickness** (por defecto 2) son píxeles lógicos. **ShowDescriptions** oculta o muestra la segunda línea de las etiquetas. Los textos de las etiquetas que no caben se recortan al espacio disponible.



StepperPosition de un asistente

Un asistente coloca su banda con **StepperPosition**: wspTop (por defecto) muestra una banda horizontal en la parte superior, de 80 píxeles lógicos de alto, wspLeft una banda vertical a la izquierda, de 200 píxeles lógicos de ancho, y wspNone oculta la banda. La barra de botones de la parte inferior tiene 52 píxeles lógicos de alto.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.LayoutExample;
begin
  // the wizard: the stepper band on the left, drawn vertically
  Wizard.StepperPosition := wspLeft;

  // a standalone stepper: vertical, captions only, bigger circles
  Stepper.Align := alLeft;
  Stepper.Width := 200;
  Stepper.Orientation := woVertical;
  Stepper.ShowDescriptions := False;
  Stepper.CircleSize := 32;
```

```
  Stepper.ConnectorThickness := 3;
end;
```

Consulte también

- Estados de los pasos
- TsgcVCLStepper
- TsgcVCLWizard

Títulos de los botones y localización

Los cuatro títulos de la barra de botones son propiedades publicadas del asistente: **BackCaption**, **NextCaption**, **FinishCaption** y **CancelCaption**, con los valores por defecto Back, Next, Finish y Cancel. Un título se guarda en el archivo del formulario solo cuando difiere del valor por defecto, por lo que un formulario con los títulos por defecto adopta los nuevos valores por defecto, y los títulos se pueden traducir con las herramientas de recursos habituales del IDE o asignar por código a partir de la traducción de la aplicación. Los botones crecen con sus títulos, y **ShowCancel** oculta el botón Cancel.
Los títulos de los pasos son el Caption y la Description de las páginas, por lo que se traducen junto con las páginas.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.CaptionsExample;
begin
  // Spanish captions for the button bar
  Wizard.BackCaption := 'Anterior';
  Wizard.NextCaption := 'Siguiente';
  Wizard.FinishCaption := 'Finalizar';
  Wizard.CancelCaption := 'Cancelar';
  // no Cancel button
  Wizard.ShowCancel := False;
end;
```

Consulte también

- Finalizar y cancelar
- TsgcVCLWizard

Temas

Theme establece los colores a partir de los tokens de tema de sgcUICore: utLight (por defecto), utDark o utCustom. **AccentColor** sustituye el acento del tema por un color $AARRGGBB, usado por los pasos activos y completados, los conectores hechos y el botón por defecto. 0 (por defecto) conserva el acento del tema.
Un asistente propaga Theme y AccentColor a su banda de stepper y a su barra de botones, se pinta a sí mismo y pinta sus páginas con el color Surface del tema, y aplica el tema a los controles hijos directos de cada página: cuando cambia el tema, cuando se carga el asistente y cuando se agrega un control a una página. Un control hijo con su propia propiedad Theme recibe el tema. Cualquier otro hijo recibe el color Surface como fondo y el color Text como color de fuente, de modo que un TLabel, un TEdit o un TCheckBox sigue siendo legible en una página oscura.

- **FireMonkey**: los controles de edición con estilo, como TEdit, TMemo, TComboBox, TListBox, los controles de fecha y las cuadrículas, conservan los colores de su estilo, porque el estilo pinta un fondo opaco propio. Su color de texto solo cambia cuando también se establece su color de fondo.
- **utCustom** deja intactos los controles hijos. En VCL las páginas toman el color de su padre, en FireMonkey ni el asistente ni las páginas se pintan.

Los tokens de tema son los mismos que los de sgcDashboard y los demás productos de sgcUICore, de modo que un asistente y un dashboard con el mismo Theme coinciden, y un asistente alojado en un mosaico de sgcDashboard sigue el tema del dashboard.

Ejemplo

```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

procedure TFeaturesForm.ThemesExample;
begin
  // the wizard, its stepper band, its button bar and the page children
  Wizard.Theme := utDark;
  // a green accent, $AARRGGBB; 0 uses the accent of the theme
  Wizard.AccentColor := $FF2E7D32;
  Stepper.Theme := utDark;
end;
```

Consulte también

- Arquitectura
- Disposición del stepper
- Renderizado sin interfaz

Tiempo de diseño

El paquete de tiempo de diseño registra TsgcVCLWizard, TsgcVCLStepper, TsgcFMXWizard y TsgcFMXStepper en la página **SGC Wizard** de la paleta. Las páginas no están en la paleta. El editor de componentes del asistente y de sus páginas agrega tres verbos al menú contextual:

Verbo	Efecto
Add Page	Crea una página nueva con el nombre de su clase, con su nombre como Caption, la coloca en último lugar en el asistente, la muestra y la selecciona.
Next Page	Muestra la página siguiente en el diseñador.
Previous Page	Muestra la página anterior en el diseñador.

Un clic en un paso de la banda del stepper en el diseñador también muestra su página. Solo la página activa es visible, y la página mostrada en el diseñador es el ActivePageIndex almacenado en el archivo del formulario. La banda del stepper y la barra de botones son internas: no se almacenan en el archivo del formulario y no se pueden seleccionar. Los Steps de un stepper independiente se editan con el editor de colecciones estándar.

Consulte también

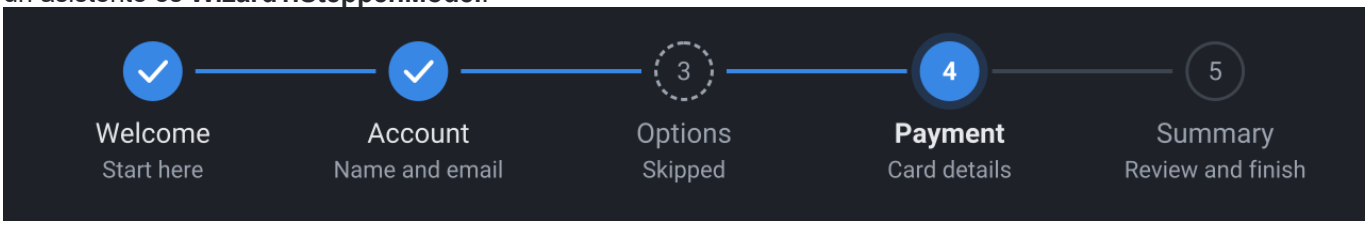
- Inicio rápido VCL
- TsgcVCLWizardPage
- Plataformas

Renderizado sin interfaz

El renderizador de sgcWizard_Render dibuja un TsgcStepperModel y la barra de botones sin ventana y sin VCL ni FMX, de modo que una aplicación de consola, un servicio, una prueba o un servidor web pueden generar el mismo stepper como imagen, por ejemplo para la documentación de una aplicación o una prueba visual. La unidad se compila en todas las plataformas, incluida Linux64.

Rutina	Descripción
sgcWizardSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	Escribe un archivo o un flujo PNG. aWidth y aHeight son píxeles lógicos y aScale los multiplica: 640 por 100 con escala 2 da una imagen de 1280 por 200.
sgcWizardSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	Escribe un archivo o un flujo SVG del mismo stepper.
sgcWizardRender(aModel, aCanvas, aWidth, aHeight, aScale = 1)	Dibuja el stepper en cualquier lienzo de sgcUICore, en píxeles de dispositivo.
sgcWizardSaveButtonBarToPNG, sgcWizardSaveButtonBarToSVG (aButtons, aStream, aWidth, aHeight, aScale = 1, aTheme = utLight, aAccent = 0)	Escribe una barra de botones. sgcWizardButton(aCaption, aEnabled, aIsDefault, aHot, aPressed) construye un botón, y un botón con el título vacío se oculta.
sgcWizardRenderButtonBar, sgcWizardButtonBarLayout, sgcWizardButtonBarHitTest	Dibujan, disponen y prueban el impacto de una barra de botones en cualquier lienzo de sgcUICore.

El modelo de un control stepper está disponible a través de su propiedad **Model**, de modo que **sgcWizardSaveToPNG(Stepper1.Model, ...)** guarda lo que muestra el control con cualquier tamaño. La banda de un asistente es **Wizard1.Stepper.Model**.



Ejemplo

```
uses
  SysUtils, Classes,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_Render;

procedure Main;
var
  oModel: TsgcStepperModel;
  oStream: TFileStream;
begin
  oModel := TsgcStepperModel.Create;
  try
    oModel.Steps.AddStep('Welcome', 'Start here');
    oModel.Steps.AddStep('Account', 'Name and email');
    oModel.Steps.AddStep('Options', 'Skipped');
    oModel.Steps.AddStep('Payment', 'Card details');
    oModel.Steps.AddStep('Summary', 'Review and finish');
    oModel.Steps[2].State := wssSkipped;
    oModel.ActiveIndex := 3;
    oModel.Theme := utDark;
```

```
    // 640 x 100 logical pixels, written at scale 2 as 1280 x 200 pixels
    sgcWizardSaveToPNG(oModel, 'wizard.png', 640, 100, 2);
    sgcWizardSaveToSVG(oModel, 'wizard.svg', 640, 100);

    // the same steps drawn vertically
    oModel.Orientation := woVertical;
    sgcWizardSaveToPNG(oModel, 'wizard_vertical.png', 220, 340);

    // a button bar: Back, Next as the default button, Cancel
    oStream := TFileStream.Create('buttons.png', fmCreate);
    try
      sgcWizardSaveButtonBarToPNG([sgcWizardButton('Back'),
        sgcWizardButton('Next', True, True), sgcWizardButton('Cancel')],
        oStream, 640, 52, 1, utDark);
    finally
      oStream.Free;
    end;
  finally
    oModel.Free;
  end;
end;
```

Consulte también

- TsgcStepperModel
- Plataformas
- Estados de los pasos

Descripción general de .NET

sgcWizard .NET es un port administrado en C# de la biblioteca Delphi sgcWizard, no un envoltorio (wrapper) alrededor de ella. Las clases C# viven en el espacio de nombres **esegece.sgcWizard** y conservan los nombres de Delphi, de modo que **TsgcStepperModel**, **TsgcWizardSteps** y **TsgcWizardStepState** se leen igual en ambos lenguajes, y el asistente, la página y el stepper de cada host siguen el patrón de nombres de Delphi. Se dibuja con el port .NET de **sgcUICore**, con el mismo canvas, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, de modo que un asistente se ve igual en WinForms, WPF, Avalonia y en la salida PNG y SVG sin interfaz.

Paquetes

El port se divide en cinco paquetes NuGet, de modo que un proyecto solo incluye el host que utiliza.

PackageId	Contiene	Frameworks de destino
esegece.sgcWizard	El modelo del stepper (**TsgcStepperModel**, **TsgcWizardSteps**, **TsgcWizardStep**), el renderizador **sgcWizard_Render** y la salida SVG sin interfaz. Sin dependencia nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcWizard.Rendering	La salida PNG sin interfaz, **sgcWizard_Render_Raster**, dibujada en el canvas SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcWizard.WinForms	**TsgcWinFormsWizard**, **TsgcWinFormsWizardPage**, **TsgcWinFormsStepper** y **TsgcWinFormsWizardButtonBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcWizard.Wpf	**TsgcWpfWizard**, **TsgcWpfWizardPage**, **TsgcWpfStepper** y **TsgcWpfWizardButtonBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcWizard.Avalonia	**TsgcAvaloniaWizard**, **TsgcAvaloniaWizardPage**, **TsgcAvaloniaStepper** y **TsgcAvaloniaWizardButtonBar**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0

Los tres paquetes de host dependen de esegece.sgcWizard.Rendering, que depende de esegece.sgcWizard, de modo que una sola línea `PackageReference` trae todo lo que un host necesita. Consulte Instalación desde NuGet.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcWizard, esegece.sgcWizard.Rendering, esegece.sgcWizard.WinForms, esegece.sgcWizard.Wpf, esegece.sgcWizard.Avalonia	Todo: los asistentes, las páginas, los steppers y las barras de botones de los tres hosts, la validación, las páginas omitidas, los estados de los pa-

		sos, los temas y la salida PNG y SVG sin interfaz.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcWizard.WinForms.Community. Dependen de los paquetes Community de sgcUICore.	El mismo contenido que Registered. Solo cambia la licencia.

La edición Community es gratuita para uso comercial y no comercial por parte de particulares y organizaciones con menos de 1.000.000 EUR de ingresos brutos anuales y no más de cinco desarrolladores. Una organización que alcance o supere cualquiera de los dos umbrales necesita una licencia Registered. El port .NET no tiene edición Trial.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (TsgcVCLWizard)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	VCL y FMX	Windows, .NET Framework 4.7.2 y .NET 8 y 9. Las páginas se declaran en XAML.
Avalonia	FMX (TsgcFMXWizard)	Windows, Linux y macOS, .NET 8 y 9. Las páginas se declaran en XAML.
Sin interfaz	sgcWizard_Render	Cualquier plataforma de los frameworks de destino: SVG con esegece.sgcWizard, PNG con esegece.sgcWizard.Rendering.

Todos los hosts tienen la misma API: las páginas, ActivePageIndex, Next, Back, GoToPage, Finish y Cancel, los eventos OnBeforePageChange, OnPageChanged, OnFinish y OnCancel, los textos de los botones, StepperPosition, ShowCancel, Theme y AccentColor. Consulte Clases de .NET para los nombres de las clases y Diferencias de .NET respecto a Delphi para lo que cambia respecto a Delphi.

Consulte también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Renderizado sin interfaz en .NET
- Demos de .NET
- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, y los paquetes core y de renderizado vienen con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcWizard.WinForms
dotnet add package esegece.sgcWizard.Wpf
dotnet add package esegece.sgcWizard.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcWizard.WinForms.Community
dotnet add package esegece.sgcWizard.Wpf.Community
dotnet add package esegece.sgcWizard.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcWizard.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los paquetes Community, nunca ambos: contienen los mismos tipos en el mismo espacio de nombres.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:sgc="clr-namespace:esegece.sgcWizard;assembly=esegece.sgcWizard.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Asigne el espacio de nombres en XAML con `xmlns:sgc="using:esegece.sgcWizard"`.
- Sin interfaz: esegece.sgcWizard por sí solo basta para SVG. Añada esegece.sgcWizard.Rendering para PNG, trae SkiaSharp y sus recursos nativos para Windows y Linux.

En C# todo host necesita `using esegece.sgcWizard;`, y `using esegece.sgcUICore;` para **TsgcUITheme** y los demás tipos de los temas.

Consulte también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos de .NET

Inicio rápido WinForms

TsgcWinFormsWizard es el asistente de WinForms, el equivalente del TsgcVCLWizard de Delphi. Es un contenedor de páginas: cada **TsgcWinFormsWizardPage** es un paso, la banda del stepper muestra los pasos y la barra de botones se mueve entre las páginas. El paquete es **esegece.sgcWizard.WinForms**.

Procedimiento

- Cree un proyecto WinForms y añada el paquete esegece.sgcWizard.WinForms.
- Cree un **TsgcWinFormsWizard**, acóplelo en el formulario y añada una página por paso con **AddPage**. Establezca la **Description** de cada página: es la descripción de su paso, y el título (caption) es el argumento de AddPage.
- Añada los controles de cada página a los **Controls** de la página.
- Asigne **OnBeforePageChange** para validar una página y establezca aAllow en false cuando no esté completa, y **OnFinish** para usar los datos.
- Ejecute: Next y Enter avanzan, Back retrocede, Finish aparece en la última página, y Cancel o Escape llaman a OnCancel.

Ejemplo

El Program.cs completo de una aplicación WinForms con un asistente de tres páginas. Compila tal cual en un proyecto `net8.0-windows` que referencia esegece.sgcWizard.WinForms.
Las páginas también se pueden crear con `new TsgcWinFormsWizardPage()` y añadir con `Controls.Add`: el orden de las páginas es el orden en que se añadieron. El asistente y el stepper están en la barra de herramientas, pero no hay verbo Add Page, consulte Diferencias de .NET respecto a Delphi.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WPF
- Inicio rápido Avalonia
- Validación

Inicio rápido WPF

TsgcWpfWizard es el asistente de WPF. Sus páginas son los elementos **TsgcWpfWizardPage** declarados en XAML como sus hijos, y cada página contiene un elemento de contenido, normalmente un panel. El paquete es **esegece.sgcWizard.Wpf**.

Procedimiento

- Cree un proyecto WPF (`dotnet new wpf`) y añada el paquete esegece.sgcWizard.Wpf.
- Asigne el espacio de nombres `esegece.sgcWizard` y declare un **TsgcWpfWizard** con un **TsgcWpfWizardPage** por paso. **Caption** y **Description** de cada página son el título y la descripción de su paso.
- Asigne los eventos en el code behind: son delegados, no eventos de WPF, por lo que no se pueden establecer como atributos XAML.

Ejemplo

MainWindow.xaml declara el asistente y sus tres páginas. El App.xaml del proyecto es el de la plantilla de WPF, con StartupUri establecido en MainWindow.xaml.
MainWindow.xaml.cs asigna los eventos y valida la página Account.
Las páginas también se pueden crear en código con `Wizard.AddPage("Caption")`, que devuelve el nuevo **TsgcWpfWizardPage**, y el asistente las expone en **Pages**.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WinForms
- Inicio rápido Avalonia
- Validación

Inicio rápido Avalonia

TsgcAvaloniaWizard es el asistente de Avalonia, el equivalente del TsgcFMXWizard de Delphi, en Windows, Linux y macOS. Sus páginas son los elementos **TsgcAvaloniaWizardPage** declarados en XAML como sus hijos, y cada página contiene un hijo. El paquete es **esegece.sgcWizard.Avalonia**.

Procedimiento

- Cree un proyecto .NET 8 y añada los paquetes esegece.sgcWizard.Avalonia, Avalonia.Desktop y Avalonia.Themes.Fluent.
- Asigne el espacio de nombres con `xmlns:sgc="using:esegece.sgcWizard"` y declare un **TsgcAvaloniaWizard** con un **TsgcAvaloniaWizardPage** por paso, con su **Caption** y su **Description**.
- Asigne los eventos en el code behind: son delegados, no eventos de Avalonia.

Ejemplo

MainWindow.axaml declara el asistente y sus tres páginas.
MainWindow.axaml.cs establece el tema, asigna los eventos y muestra un mensaje en la página Account mientras el nombre esté vacío.
Program.cs inicia la aplicación con el tema Fluent.
Las páginas también se pueden crear en código con `Wizard.AddPage("Caption")`, que devuelve el nuevo **TsgcAvaloniaWizardPage**; establezca su **Child** con el contenido de la página.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WinForms
- Inicio rápido WPF
- Validación

Clases de .NET

Todo host tiene un asistente, una página, un stepper y una barra de botones con los mismos miembros. Los tipos independientes del framework son compartidos por los tres hosts y conservan sus nombres de Delphi.

Clases de los hosts

Función	WinForms	WPF	Avalonia
Asistente	TsgcWinFormsWizard	TsgcWpfWizard	TsgcAvaloniaWizard
Página	TsgcWinFormsWizardPage (un contenedor de Control)	TsgcWpfWizardPage (un elemento Content)	TsgcAvaloniaWizardPage (un Border con un Child)
Colección de páginas	ReadOnlyCollection<TsgcWinFormsWizardPage>	TsgcWpfWizardPages	AvaloniaList<TsgcAvaloniaWizardPage>
Stepper	TsgcWinFormsStepper	TsgcWpfStepper	TsgcAvaloniaStepper
Barra de botones	TsgcWinFormsWizardButtonBar	TsgcWpfWizardButtonBar	TsgcAvaloniaWizardButtonBar

Tipos independientes del framework

Tipo	Descripción
TsgcStepperModel	Los pasos, el paso activo, la orientación, los estados automáticos, el diseño y la prueba de impacto (hit test). El stepper de cada host posee uno, consulte TsgcStepperModel.
TsgcWizardSteps, TsgcWizardStep	La colección de pasos y un paso: Caption, Description, Glyph, State, Enabled, Visible y Tag.
TsgcWizardStepState	wssPending, wssActive, wssCompleted, wssError, wssSkipped.
TsgcWizardOrientation	woHorizontal, woVertical.
TsgcWizardStepperPosition	wspTop, wspLeft, wspNone.

TsgcWizardPageChangeEvent	`void (object Sender, int aFromIndex, int aToIndex, ref bool aAllow)`
TsgcWizardStepClickEvent	`void (object Sender, int aIndex, ref bool aAllow)`
TsgcUINotifyEvent	`void (object Sender)`, de esegece.sgcUICore.
sgcWizard_Render, sgcWizard_Render_Raster	El renderizador y la salida SVG y PNG sin interfaz, consulte Renderizado sin interfaz en .NET.

Asistente

Miembro	Tipo	Descripción
Pages	colección	Las páginas en orden. PageCount e IndexOfPage(aPage) como en Delphi.
AddPage(string aCaption)	página	Crea una página, la añade al final y la devuelve.
ActivePageIndex, ActivePage	int, página	La página visible. Al establecerlos se cambia de página directamente, sin eventos.
Next(), Back(), GoToPage(int aIndex)	bool	Pasa a otra página a través de OnBeforePageChange, omitiendo las páginas cuyo Skip es true. Next en la última página llama a Finish.
Finish(), Cancel()	bool, void	Finish llama a OnBeforePageChange con aToIndex = PageCount y después a OnFinish. Cancel llama a OnCancel.
NextPageIndex(int), PrevPageIndex(int), IsLastPage()	int, bool	La página siguiente y la anterior que no se omiten, y si la página activa es la última.
StepperPosition	TsgcWizardStepperPosition	Dónde está la banda del stepper: arriba, a la izquierda u oculta.
BackCaption, NextCaption, FinishCaption, CancelCaption	string	Los textos de los botones, para la localización.
ShowCancel	bool	Muestra el botón Cancel.
Theme, AccentColor	TsgcUITheme, uint	El tema del asistente, de su banda del stepper, de su barra de botones y de los controles de sus páginas. AccentColor 0 usa el acento del tema.
Stepper, ButtonBar	stepper, barra de botones	La banda del stepper y la barra de botones internas.
OnBeforePageChange	TsgcWizardPageChangeEvent	Antes de cada cambio de página y antes de Finish. aAllow false lo veta.
OnPageChanged, OnFinish, OnCancel	TsgcUINotifyEvent	Después de un cambio de página, en Finish y en Cancel.

Página

Miembro	Tipo	Descripción
Caption	string	El título (caption) del paso. En WinForms es el Text del control.
Descripción	string	La descripción del paso y su información sobre herramientas (tooltip).
StepGlyph	string	El texto dentro del círculo del paso en lugar de su número.
Skip	bool	Next, Back y GoToPage saltan la página y el paso se dibuja como omitido.
Wizard, PageIndex	asistente, int	El asistente de la página y su índice.
Content (WPF), Child (Avalonia), Controls (WinForms)	elemento	Lo que muestra la página.

Stepper

El stepper funciona por sí solo, sin un asistente. Sus miembros son los de los steppers de Delphi: Model, Steps, Orientation, ActiveIndex, AutoStates, ShowDescriptions, CircleSize, ConnectorThickness, Theme, AccentColor, Clickable, StepAt(X, Y), StepRect(aIndex), OnStepClick (TsgcWizardStepClickEvent, aAllow false veta el clic) y OnActiveChange, que se dispara solo por interacción del usuario. Wizard es el asistente que lo posee, o null para un stepper independiente. Consulte TsgcVCLStepper.

Barra de botones

La barra de botones de un asistente dibuja Back, Next (Finish en la última página) y Cancel. GetButtons(), ButtonCount(), Buttons(aIndex), ButtonRect(aIndex) y ButtonAt(X, Y) describen los botones, y Wizard es su asistente. La crea el asistente y se accede a ella mediante su propiedad ButtonBar.
Las posiciones y los tamaños son píxeles en WinForms (Rectangle) y unidades independientes del dispositivo en WPF y Avalonia (Rect).

Consulte también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia

Diferencias de .NET respecto a Delphi

La API de .NET sigue la API de Delphi nombre por nombre. Estos son los puntos en que difiere.

Eventos

OnBeforePageChange, OnPageChanged, OnFinish, OnCancel, OnStepClick y OnActiveChange son propiedades delegado con los nombres de Delphi, no miembros `event` de C#. Asignar un manejador reemplaza al anterior, como un puntero a método de Delphi, y se asignan en código, también en WPF y Avalonia, donde XAML no puede establecerlos. El parámetro `var aAllow: Boolean` de Delphi es `ref bool aAllow`, y un manejador de OnBeforePageChange tiene la firma `void (object Sender, int aFromIndex, int aToIndex, ref bool aAllow)`. Los inicios rápidos muestran los manejadores de los tres hosts.

GoToPage y navegación

GoToPage, Next, Back y Finish devuelven bool y se comportan como en Delphi: GoToPage devuelve false cuando el índice está fuera de rango o la página se omite, devuelve true cuando la página ya está activa, y en caso contrario pasa por OnBeforePageChange. Finish llama a OnBeforePageChange con aToIndex = PageCount, de modo que aToIndex > aFromIndex es todo avance. Un clic en un paso completado de la banda del stepper llama a GoToPage. Establecer ActivePageIndex o ActivePage cambia la página sin eventos, como en Delphi.

La colección de páginas

Delphi tiene la propiedad indexada `Pages[Index]` y `PageCount`. En .NET **Pages** es una colección, por lo que `Pages[i]`, `Pages.Count` y `foreach` funcionan como de costumbre:

- WinForms: una colección de solo lectura. Una página se añade con AddPage o añadiendo un TsgcWinFormsWizardPage a los Controls del asistente, y el orden de las páginas es el orden en que se añadieron, no el orden z de los controles.
- WPF: TsgcWpfWizardPages, la propiedad de contenido del asistente, por lo que las páginas son los hijos del asistente en XAML. Cada página tiene un elemento Content.
- Avalonia: una AvaloniaList, la propiedad de contenido del asistente. Cada página es un Border con un Child.

Sugerencias y tooltips

En Delphi la sugerencia de un paso es su Description, que se muestra cuando ShowHint es True, y FMX muestra sugerencias desde 10.1 Berlin. En .NET el stepper siempre muestra la Description del paso bajo el ratón como tooltip, sin propiedad ShowHint: WinForms usa un componente ToolTip interno, WPF establece la propiedad ToolTip del stepper y Avalonia establece la propiedad adjunta ToolTip.Tip. No establezca usted mismo el tooltip de un stepper, el stepper lo reemplaza; establezca en su lugar la Description del paso o de la página.

Sin verbos del diseñador

El editor de componentes de Delphi tiene los verbos Add Page, Next Page y Previous Page y muestra otra página en el diseñador cuando se hace clic en un paso, consulte Tiempo de diseño. El port .NET no tiene verbos del diseñador. En WinForms el asistente y el stepper están en la barra de herramientas, y las páginas se crean en código con AddPage. En WPF y Avalonia las páginas se declaran en XAML.

Teclado

Enter es Next o Finish y Escape es Cancel mientras el foco está dentro del asistente, como en Delphi. En WinForms las teclas llegan a través de ProcessDialogKey, por lo que un TextBox multilínea o un botón con foco las conserva. En WPF y Avalonia llegan a través del evento KeyDown en burbuja (bubbling), por lo que un control que gestiona la tecla tiene prioridad.

Ediciones

El port .NET se distribuye en paquetes Registered y Community con el mismo contenido, consulte Descripción general de .NET. No hay edición Trial ni interruptor de compilación en el código.

Enlace de datos

En WPF y Avalonia las propiedades visibles para el usuario son propiedades de dependencia y propiedades de Avalonia, por lo que se pueden enlazar, aplicar con estilos y animar, consulte Enlace de datos en .NET.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia

Enlace de datos en .NET

En los hosts WPF y Avalonia, las propiedades visibles para el usuario del asistente, de sus páginas y del stepper son propiedades de dependencia (WPF) y propiedades de Avalonia, por lo que se pueden enlazar a un modelo de vista, establecer mediante un Style y animar. Este tema se aplica solo a WPF y Avalonia. Los controles WinForms no cambian: sus propiedades son propiedades CLR simples, como antes.

Propiedades enlazables

Cada propiedad de la tabla tiene un campo estático con el sufijo Property, por ejemplo `TsgcWpfWizard.ActivePageIndexProperty`. En WPF los campos son DependencyProperty. En Avalonia son StyledProperty, excepto ActivePageIndex y ActiveIndex, que son DirectProperty. Las propiedades marcadas con **TwoWay** se enlazan TwoWay de forma predeterminada, porque el propio control las modifica (Next, Back, GoToPage, un clic en un paso), de modo que un enlace sin Mode mantiene actualizado el modelo de vista. Las demás propiedades se enlazan OneWay de forma predeterminada.

Clase WPF	Clase Avalonia	Propiedades enlazables
TsgcWpfWizard	TsgcAvaloniaWizard	**ActivePageIndex (TwoWay)**, StepperPosition, BackCaption, NextCaption, FinishCaption, CancelCaption, ShowCancel, Theme, AccentColor
TsgcWpfWizardPage	TsgcAvaloniaWizardPage	Caption, Description, StepGlyph, Skip
TsgcWpfStepper	TsgcAvaloniaStepper	Orientation, **ActiveIndex (TwoWay)**, AutoStates, ShowDescriptions, CircleSize, ConnectorThickness, Theme, AccentColor, Clickable

En Avalonia, Theme es el tema de sgcUICore (TsgcUITheme) y oculta el tema de control de Avalonia de StyledElement, que sigue accesible mediante `StyledElement.ThemeProperty`.

Comportamiento

- **Un enlace navega, el código no.** Un nuevo ActivePageIndex que llega a través de un enlace de datos es una navegación: pasa por OnBeforePageChange y después por OnPageChanged, como GoToPage. Establecido desde el código (el setter CLR), ActivePageIndex cambia de página sin eventos, como en Delphi.
- **Un veto se escribe de vuelta.** Cuando OnBeforePageChange establece aAllow en false, el asistente conserva la página actual y restaura el índice anterior en el control y en el modelo de vista enlazado. En WPF el modelo de vista recibe el índice anterior cuando el dispatcher ha ejecutado el trabajo pendiente.
- **Un índice fuera de rango se ignora.** Un índice enlazado que no es una página mantiene la página actual y no genera ningún evento, y un origen TwoWay recibe de vuelta el índice en uso, en WPF y en Avalonia. El enlace sigue activo, así que el siguiente índice válido navega.
- **Un valor corregido llega al origen.** Un valor que el control corrige, como un CircleSize o un ActiveIndex que el stepper limita, se escribe de vuelta en la propiedad y en un origen TwoWay, en WPF y en Avalonia. En WPF el origen lo recibe cuando el dispatcher ha ejecutado el trabajo pendiente. En Avalonia la escritura de vuelta cubre un valor que el origen envía cuando el enlace ya está en marcha. Un valor que el origen ya tiene cuando el enlace se inicia se corrige en el control, pero no se escribe de vuelta en el origen.
- **El modelo de vista sigue al asistente.** Next, Back, GoToPage, un clic en un paso y un cambio desde el código copian el nuevo índice en ActivePageIndex, y un enlace TwoWay lo pasa al modelo de vista.
- **Styles.** Un setter de Style puede establecer cualquier propiedad enlazable, por ejemplo el Theme y las opciones del stepper de todos los asistentes de una ventana.

Propiedades que siguen siendo propiedades CLR

Propiedad	Motivo
Pages (asistente)	La colección de páginas es la propiedad de contenido del asistente, rellenada por las páginas declaradas en XAML, no un valor que enlazar.
ActivePage (asistente)	Una referencia a un elemento de página. Enlace ActivePageIndex en su lugar, que sigue cada cambio de página.
PageIndex (página)	Solo lectura, la posición de la página en Pages.
Steps (stepper)	La colección de pasos. El asistente rellena los pasos de su propio stepper a partir de las páginas.
OnBeforePageChange, OnPageChanged, On-Finish, OnCancel, OnStepClick, OnActive-Change	Propiedades delegado con los nombres de Delphi, asignadas en código, consulte Diferencias de .NET respecto a Delphi.

Ejemplo

El modelo de vista siguiente se comparte en ambos ejemplos. PageIndex está enlazado a ActivePageIndex sin Mode, por lo que se enlaza TwoWay, NextCaption está enlazado al título del botón Next, y SkipOptions está enlazado a la propiedad Skip de la página Options y a una casilla de verificación.

WPF

MainWindow.xaml crea el modelo de vista como DataContext de la ventana. El código subyacente y App.xaml son los de la plantilla WPF.

Avalonia

MainWindow.axaml establece `x:DataType` con el modelo de vista, por lo que los enlaces son enlaces compilados. El código subyacente llama a InitializeComponent, y la clase App es la de Inicio rápido Avalonia.

Consulte también

- Diferencias de .NET respecto a Delphi
- Clases de .NET
- Inicio rápido WPF
- Inicio rápido Avalonia

Renderizado sin interfaz en .NET

Un stepper se puede dibujar sin ventana, en un servidor, en una prueba o en un informe. **sgcWizard_Render.sgcWizardSaveToSVG** escribe un archivo o flujo SVG solo con el paquete esegece.sgcWizard, y **sgcWizard_Render_Raster.sgcWizardSaveToPNG** escribe un PNG con esegece.sgcWizard.Rendering, que dibuja sobre SkiaSharp en Windows, Linux y macOS. La salida SVG es idéntica byte a byte a la del renderizado sin interfaz de Delphi.

Métodos

Método	Paquete	Descripción
sgcWizard_Render.sgcWizardSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	esegece.sgcWizard	Escribe el stepper de aModel como SVG.
sgcWizard_Render_Raster.sgcWizardSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	esegece.sgcWizard.Rendering	Escribe el stepper de aModel como PNG sobre fondo transparente.
sgcWizard_Render.sgcWizardSaveButtonBarToSVG(aButtons, aStream, aWidth, aHeight, aScale, aTheme, aAccent)	esegece.sgcWizard	Escribe una barra de botones como SVG.
sgcWizard_Render_Raster.sgcWizardSaveButtonBarToPNG(aButtons, aStream, aWidth, aHeight, aScale, aTheme, aAccent)	esegece.sgcWizard.Rendering	Escribe una barra de botones como PNG.

sgcWizard_Render.sgcWizardButton(aCaption, aEnabled = true, aIsDefault = false, aHot = false, aPressed = false)	esegece.sgcWizard	Devuelve un TsgcWizardButton para una barra de botones.
sgcWizard_Render.sgcWizardRender(aModel, aCanvas, aWidth, aHeight, aScale = 1)	esegece.sgcWizard	Dibuja el stepper en cualquier canvas de sgcUICore.

aWidth y aHeight son píxeles lógicos. La imagen mide aWidth x aScale por aHeight x aScale píxeles, de modo que una escala de 2 da una imagen nítida para una pantalla de alta densidad o para imprimir.

Ejemplo

Un programa de consola (`net8.0`) que referencia esegece.sgcWizard.Rendering y escribe un stepper oscuro con un paso omitido como PNG y SVG, los mismos pasos dibujados en vertical y una barra de botones.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Renderizado sin interfaz
- TsgcStepperModel

Demos de .NET

Las demos de .NET están en la carpeta `demos` del producto y en el zip de demos, con la solución `sgcWizard_Demos.sln`. Cada demo tiene un README.md.

Demo	Host	Descripción
Onboarding	WinForms	Un asistente de cuenta de cuatro páginas (Welcome, Account, Options y Summary) sobre un TsgcWinFormsWizard. OnBeforePageChange valida el nombre y el correo electrónico de la página Account, una casilla de verificación establece Skip en la página Options, OnPageChanged rellena el resumen, OnFinish lo muestra y OnCancel pregunta antes de cerrar. La barra de herramientas cambia el tema y coloca la banda del stepper arriba, a la izquierda o la oculta.
OnboardingWpf	WPF	El mismo asistente sobre un TsgcWpfWizard declarado en MainWindow.xaml.
OnboardingAvalonia	Avalonia	El mismo asistente sobre un TsgcAvaloniaWizard, en Windows, Linux y macOS.
Stepper	WinForms	Dos controles TsgcWinFormsStepper independientes, uno horizontal y otro vertical, con los cinco pasos de una compra, mantenidos sincronizados. Los botones avanzan, retroceden, marcan el paso activo como error, lo omiten y reinician los pasos, y un clic en un paso completado lo activa a través de OnStepClick.
StepperWpf	WPF	Los mismos steppers como controles TsgcWpfStepper colocados en MainWindow.xaml.
StepperAvalonia	Avalonia	Los mismos steppers como controles TsgcAvaloniaStepper.

Compilar y ejecutar

Las demos necesitan el SDK de .NET 8 o posterior. Ejecute una demo desde su carpeta:

```
dotnet run
```

Dentro del árbol del producto las demos se compilan contra los proyectos de la biblioteca. Desde el zip de demos restauran los paquetes NuGet, la edición Community por defecto. Establezca **SgcWizardEdition** para usar los paquetes Registered:

```
dotnet build -p:SgcWizardEdition=Registered
```

Autocomprobación

Cada demo tiene un modificador `--test` que abre la ventana fuera de pantalla, maneja el asistente o los steppers, escribe una línea PASS o FAIL por comprobación en el archivo y termina con el número de comprobaciones fallidas, de modo que 0 significa que todas pasaron.

```
dotnet run -- --test result.txt
```

Consulte también

- Descripción general de .NET
- Instalación desde NuGet

- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos

Solución de problemas

EsgcWizardError: A TsgcVCLWizardPage can only be placed on a TsgcVCLWizard

Una página debe ser hija directa de un asistente. Asignar el Parent de un TsgcVCLWizardPage a un panel, un formulario o cualquier otro control genera EsgcWizardError, y lo mismo ocurre con un TsgcFMXWizardPage fuera de un TsgcFMXWizard. Cree las páginas con **AddPage**, o asigne el asistente a su Parent. Para agrupar controles en una página, coloque un TPanel o un marco en la página. Consulte TsgcVCLWizardPage.

Un programa de prueba pide derechos de administrador

Windows trata un ejecutable cuyo nombre contiene palabras como setup, install o update como un instalador y pide elevación antes de iniciarlo. Cuando pruebe un asistente, no llame al programa SetupWizard.exe ni InstallWizard.exe, elija un nombre como Onboarding.exe, o agregue un manifiesto con requestedExecutionLevel asInvoker.

Next no hace nada

OnBeforePageChange asignó False a aAllow, que es como la validación veta un cambio, o las páginas siguientes están todas omitidas y Next llama a Finish. Compruebe el controlador y la propiedad Skip de las páginas. Consulte Validación.

Finish no cierra el formulario

El asistente nunca cierra nada. Cierre el formulario o asigne ModalResult en OnFinish, y en OnCancel. Consulte Finalizar y cancelar.

Enter no pasa a la página siguiente

Un control que quiere Enter se lo queda: un TMemo, un botón con foco, o un TButton del formulario con Default en True. Consulte Teclado.

Un clic en un paso no cambia la página

Solo se puede hacer clic en los pasos completados, y en los pasos con error anteriores al activo, y solo mientras Clickable es True. Un clic también pasa por OnBeforePageChange.

Un campo de edición FMX conserva sus colores claros en una página oscura

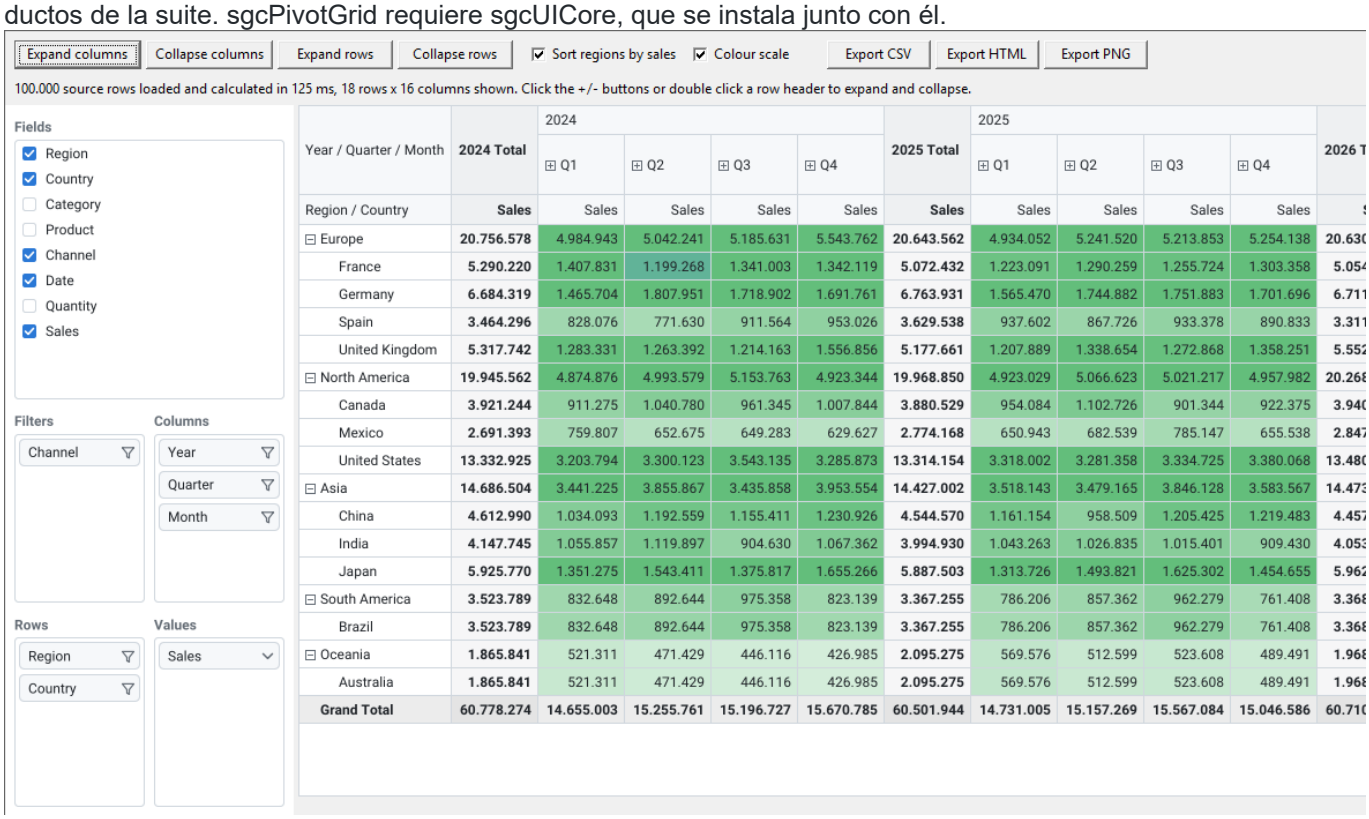
Los controles de edición de FireMonkey con estilo conservan los colores de su estilo. Use un estilo oscuro para el formulario, o establezca los colores del control. Consulte Temas.

Faltan los controles FMX

Los controles de FireMonkey necesitan Delphi XE6 o posterior, y los hints de los controles FMX necesitan 10.1 Berlin o posterior. Consulte Plataformas.

Introducción

sgcPivotGrid añade tablas dinámicas a las aplicaciones Delphi y C++Builder, para VCL y FireMonkey. **TsgcVCLPivotGrid** y **TsgcFMXPivotGrid** resumen filas de origen planas según los campos que se colocan en las filas, las columnas, los filtros y los valores: sumas, recuentos, promedios y más, con subtotales y totales generales, agrupación de fechas por año, trimestre, mes, semana o día, filtros, ordenación, expandir y contraer, drill-down hasta las filas de origen y exportación a CSV, JSON, HTML, PNG y SVG. **TsgcVCLPivotFieldList** y **TsgcFMXPivotFieldList** permiten al usuario construir el diseño con arrastrar y soltar.
Los datos proceden de un **TDataSet** mediante **TsgcPivotSource_DataSet**, de cualquier **IsgcUITable**, como un sgcGrid, mediante **TsgcPivotSource_Table**, o de su propio código con AppendRow. El cálculo lo realiza **TsgcPivotEngine**, una clase no visual con un almacén de columnas tipado, de modo que la misma tabla dinámica también funciona sin ventana, en un servidor o en Linux64, y **TsgcPivotGridRenderer** la dibuja en PNG y SVG. Un millón de filas de origen se recalculan en una fracción de segundo.
sgcPivotGrid forma parte de la suite visual de eSeGeCe. Se dibuja con **sgcUICore**, el paquete de ejecución compartido con el lienzo, el rasterizador, las fuentes, el escritor SVG y los temas claro y oscuro, de modo que una tabla dinámica tiene el mismo aspecto en VCL, en FireMonkey y en la salida sin interfaz, y coincide con los demás productos de la suite. sgcPivotGrid requiere sgcUICore, que se instala junto con él.



Year / Quarter / Month	2024 Total	2024				2025 Total	2025				2026 T
		⊞ Q1	⊞ Q2	⊞ Q3	⊞ Q4		⊞ Q1	⊞ Q2	⊞ Q3	⊞ Q4	
Region / Country	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	Sales	
⊟ Europe	20.756.578	4.984.943	5.042.241	5.185.631	5.543.762	20.643.562	4.934.052	5.241.520	5.213.853	5.254.138	20.630
France	5.290.220	1.407.831	1.199.268	1.341.003	1.342.119	5.072.432	1.223.091	1.290.259	1.255.724	1.303.358	5.054
Germany	6.684.319	1.465.704	1.807.951	1.718.902	1.691.761	6.763.931	1.565.470	1.744.882	1.751.883	1.701.696	6.711
Spain	3.464.296	828.076	771.630	911.564	953.026	3.629.538	937.602	867.726	933.378	890.833	3.311
United Kingdom	5.317.742	1.283.331	1.263.392	1.214.163	1.556.856	5.177.661	1.207.889	1.338.654	1.272.868	1.358.251	5.552
⊟ North America	19.945.562	4.874.876	4.993.579	5.153.763	4.923.344	19.968.850	4.923.029	5.066.623	5.021.217	4.957.982	20.268
Canada	3.921.244	911.275	1.040.780	961.345	1.007.844	3.880.529	954.084	1.102.726	901.344	922.375	3.940
Mexico	2.691.393	759.807	652.675	649.283	629.627	2.774.168	650.943	682.539	785.147	655.538	2.847
United States	13.332.925	3.203.794	3.300.123	3.543.135	3.285.873	13.314.154	3.318.002	3.281.358	3.334.725	3.380.068	13.480
⊟ Asia	14.686.504	3.441.225	3.855.867	3.435.858	3.953.554	14.427.002	3.518.143	3.479.165	3.846.128	3.583.567	14.473
China	4.612.990	1.034.093	1.192.559	1.155.411	1.230.926	4.544.570	1.161.154	958.509	1.205.425	1.219.483	4.457
India	4.147.745	1.055.857	1.119.897	904.630	1.067.362	3.994.930	1.043.263	1.026.835	1.015.401	909.430	4.053
Japan	5.925.770	1.351.275	1.543.411	1.375.817	1.655.266	5.887.503	1.313.726	1.493.821	1.625.302	1.454.655	5.962
⊟ South America	3.523.789	832.648	892.644	975.358	823.139	3.367.255	786.206	857.362	962.279	761.408	3.368
Brazil	3.523.789	832.648	892.644	975.358	823.139	3.367.255	786.206	857.362	962.279	761.408	3.368
⊟ Oceania	1.865.841	521.311	471.429	446.116	426.985	2.095.275	569.576	512.599	523.608	489.491	1.968
Australia	1.865.841	521.311	471.429	446.116	426.985	2.095.275	569.576	512.599	523.608	489.491	1.968
Grand Total	60.778.274	14.655.003	15.255.761	15.196.727	15.670.785	60.501.944	14.731.005	15.157.269	15.567.084	15.046.586	60.710

Ediciones

sgcPivotGrid se distribuye como Registered, Trial y Community. La edición Community omite cuatro funciones avanzadas: campos calculados, mostrar valores como, Top N y persistencia del diseño. Consulte Ediciones.

Por dónde empezar

- Arquitectura: los orígenes, el almacén de columnas, el flujo del motor, la instantánea y las vistas.
- Plataformas: las versiones de Delphi, C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX, Inicio rápido C++Builder, Inicio rápido DataSet e Inicio rápido sin interfaz: una primera tabla dinámica en pocos minutos.
- Componentes: las cuadrículas, las listas de campos, el motor y sus campos, los orígenes, el renderizador y las reglas de formato.
- Áreas, Agregaciones, Agrupación de fechas, Totales, Filtros, Top N, Mostrar valores como, Campos calculados, Persistencia del diseño y Exportación.

- Expandir y contraer, Ordenación, Drill-down, Formato condicional, Interoperabilidad con gráficos, Teclado y accesibilidad y Rendimiento.
- Demos: los proyectos de demostración de Delphi y C++Builder.
- Solución de problemas: problemas habituales y sus causas.

Arquitectura

sgcPivotGrid separa el cálculo del dibujo, y el dibujo del framework. El motor y el renderizador usan solo la RTL y sgcUICore, y los controles VCL y FMX son anfitriones delgados que los rodean.

Flujo de datos

1. **Orígenes.** TsgcPivotSource_DataSet lee un TDataSet, TsgcPivotSource_Table lee cualquier IsgcUITable, y su código puede llamar directamente a AppendRow de TsgcPivotEngine. Todos escriben en el mismo almacén de columnas.
2. **Almacén de columnas.** Las columnas de texto se codifican con diccionario. Los números, las fechas y los booleanos son valores Double, con NaN como valor vacío. El tipo de una columna procede del tipo de campo DB, de DefineColumn o de su primer valor que no esté vacío, y una columna de enteros se convierte en columna de coma flotante cuando llega un valor fraccionario.
3. **Motor.** Cada recálculo codifica los miembros de los campos de fila, columna y filtro, aplica los filtros, agrega las celdas hoja en una pasada sobre las filas de origen, aplica los filtros de valor y Top N, acumula las hojas en los subtotales y totales generales con combinaciones exactas, ordena los miembros y aplana los ejes con su estado expandido. Los totales se calculan a partir de los valores de origen, nunca como promedios de promedios.
4. **Instantánea.** El resultado se publica como un TsgcPivotResult inmutable. Los lectores lo toman entre Lock y Unlock, de modo que una vista siempre dibuja un resultado completo. Expandir o contraer un miembro construye una nueva instantánea sobre los mismos datos, sin volver a agregar.
5. **Salida.** TsgcPivotGridRenderer dibuja una instantánea en cualquier lienzo de sgcUICore: la pantalla, un PNG o un SVG. TsgcVCLPivotGrid y TsgcFMXPivotGrid alojan el renderizador con desplazamiento, selección, teclado y ratón. Los métodos de exportación escriben la vista como CSV, HTML o JSON, y ResultTable la expone como un IsgcUITable, por ejemplo a sgcCharts.

Clases

Clase	Unidad	Función
TsgcVCLPivotGrid	sgcPivotGrid_VCL	La tabla dinámica VCL. Posee un motor, publicado como Pivot.
TsgcFMXPivotGrid	sgcPivotGrid_FMX	La tabla dinámica de FireMonkey.
TsgcVCLPivotFieldList	sgcPivotGrid_FieldList_VCL	La lista de campos VCL, enlazada a una cuadrícula o a un motor.
TsgcFMXPivotFieldList	sgcPivotGrid_FieldList_FMX	La lista de campos de FireMonkey.
TsgcPivotEngine	sgcPivotGrid_Engine	El almacén de columnas, el diseño, el cálculo, la instantánea, el drill-down y la exportación.
TsgcPivotField	sgcPivotGrid_Fields	Un campo del diseño: su área, agrupación, agregación, ordenación y filtros.
TsgcPivotCalculatedField	sgcPivotGrid_Fields	Una columna calculada por fila de origen a partir de una expresión.
TsgcPivotSource_DataSet	sgcPivotGrid_Source_DataSet	Carga un TDataSet y lo recarga cuando cambian los datos.
TsgcPivotSource_Table	sgcPivotGrid_Source_Table	Carga cualquier IsgcUITable y lo recarga cuando cambia su versión.

TsgcPivotGridRenderer	sgcPivotGrid_Render	Dibuja una instantánea en un lienzo, PNG o SVG, y realiza la prueba de impacto.
TsgcPivotFormatRule	sgcPivotGrid_Format	Una regla de formato condicional: una regla de valor o una escala de color.

Hilos

Un recálculo se ejecuta en el hilo que lo dispara, normalmente el hilo principal, y OnCalculated se genera en ese hilo. La instantánea está protegida por un bloqueo, de modo que el código de otro hilo lee Result solo entre Lock y Unlock. TsgcPivotSource_Table recarga en el hilo que notifica el cambio de su tabla.

Rendimiento

El grupo de rendimiento de QA mide estos escenarios con una compilación de release: una ejecución de calentamiento y después la mediana de cinco ejecuciones, con datos de una semilla fija para que cada ejecución sea idéntica. La tabla indica el presupuesto de cada escenario.

Escenario	Win64	Win32
Recalcular 1.000.000 de filas, 2 campos de fila (50 y 20 miembros), 1 campo de columna (12 miembros), 1 Sum	100 ms	160 ms
AppendRow de 1.000.000 de filas con 3 valores de texto, 1 de coma flotante y 1 de fecha	900 ms	1,400 ms
El primer diseño con CountDistinct sobre una columna de 100.000 valores	400 ms	600 ms
El primer diseño con StdDev, Average y Max como tres medidas	180 ms	280 ms
Un fotograma a 1920x1080 de 10.000 filas y 50 columnas de datos, desplazado en cada fotograma	16 ms	25 ms
Expandir o contraer un miembro de un eje con 10.000 elementos	5 ms	8 ms
Memoria del motor por encima del almacén para el primer escenario	32 MB	32 MB

Véase también

- Plataformas
- TsgcPivotEngine
- Inicio rápido sin interfaz

Ediciones

sgcPivotGrid se compila a partir del mismo código fuente para todas las ediciones.

Edición	Incluye
Registered	Todo: las cuadrículas y listas de campos VCL y FMX, el motor, ambos orígenes, todas las agregaciones, agrupación de fechas, totales, filtros, ordenación, drill-down, reglas de formato, exportación, la salida PNG y SVG sin interfaz, y las cuatro funciones avanzadas: campos calculados, mostrar valores como, Top N y persistencia del diseño.
Trial	Las mismas funciones que Registered. Una compilación con SGC_TRIAL muestra un mensaje de que es una versión de prueba cuando se crea un componente después de que haya pasado la ventana de publicación indicada en la compilación. Nada más cambia.
Community	Todo excepto las cuatro funciones avanzadas. CountDistinct, StdDev, Var, la agrupación de fechas, los filtros, el drill-down, la exportación, las reglas de formato, las listas de campos y las vistas forman parte de Community, y no hay aviso de prueba.

Las funciones avanzadas

Las funciones avanzadas se compilan bajo el define SGC_PIVOTGRID_ADVANCED, que sgcVer.inc desactiva en la edición Community. Sus unidades, sgcPivotGrid_Expr, sgcPivotGrid_ShowAs y sgcPivotGrid_Layout, no forman parte de la distribución Community, y estos miembros no existen en una compilación Community:

Función	Miembros
Campos calculados	TsgcPivotEngine.CalculatedFields, TsgcPivotCalculatedField, TsgcPivotCalculatedFields
Mostrar valores como	TsgcPivotField.ShowAs, ShowAsAxis y ShowAsBaseItem
Top N	TsgcPivotField.TopN y TopNOthers, TsgcPivotEngine.OthersCaption
Persistencia del diseño	SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile y LoadLayoutFromFile del motor y de ambas cuadrículas

La edición es una elección en tiempo de compilación en sgcVer.inc. SGC_COMMUNITY también desactiva SGC_TRIAL. El código que usa un miembro avanzado y debe compilar también contra Community lo coloca dentro de {$IFDEF SGC_PIVOTGRID_ADVANCED}. Un formulario guardado por una compilación Registered con propiedades avanzadas establecidas no se carga en una compilación Community.

Véase también

- Plataformas
- Introducción

Instalación

sgcPivotGrid forma parte de la distribución sgcUI. Un único instalador cubre los productos sgcUI, e instala solo los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcPivotGrid** y cualquier otro producto que desee. El servidor de licencias devuelve los productos que posee, y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos, después compila los paquetes en su equipo para cada IDE seleccionado, en orden de dependencias, e instala los paquetes de tiempo de diseño.

Qué incluye sgcPivotGrid

sgcUICore, la base compartida de los productos visuales de sgcUI, viene con sgcPivotGrid. No tiene casilla propia. El instalador lo instala automáticamente siempre que se selecciona sgcPivotGrid.

Carpetas instaladas

Cada producto se instala en una sola carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de todos los productos instalados. Es la única ruta de biblioteca de sgcUI.
`Packages`	Los paquetes de todos los productos instalados, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\PivotGrid`	Las demos de sgcPivotGrid. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de biblioteca y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore está en la misma carpeta `Source`, por lo que no necesita una ruta propia.

Trial y Community

- **Trial**: el instalador de prueba de sgcUI instala todos los productos en modo de prueba, sgcPivotGrid incluido.
- **Community**: el instalador gratuito sgcUI Community incluye sgcPivotGrid sin sus cuatro funciones avanzadas. Consulte Ediciones.

Compilación manual de los paquetes

Para compilar usted mismo los paquetes, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de la versión de Delphi, por ejemplo `sgcPivotGridD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (tiempo de ejecución).
`sgcPivotGrid<V>` / `dclsgcPivotGrid<V>`	Los componentes de sgcPivotGrid (tiempo de ejecución / tiempo de diseño).

Compílelos en orden de dependencias: `sgcUICore`, después `sgcPivotGrid`, después `dclsgcPivotGrid`. A continuación instale el paquete de tiempo de diseño `dclsgcPivotGrid` y añada `sgcUI\Source` a la ruta de biblioteca de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` que corresponda a su versión, por ejemplo `sgcPivotGridC13.cbproj`, y compílelo. Los encabezados `.hpp` generados aparecen junto al paquete compilado.

Véase también

- Plataformas
- Inicio rápido VCL

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13. Los paquetes generan los encabezados .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	TsgcVCLPivotGrid y TsgcVCLPivotFieldList en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	TsgcFMXPivotGrid y TsgcFMXPivotFieldList desde Delphi XE6 (SGC_PIVOTGRID_FMX). Usan solo FireMonkey y sgcUICore, con sus propias barras de desplazamiento y gesto táctil de desplazamiento.
Linux64	Las unidades independientes del framework desde Delphi 10.2 (SGC_PIVOTGRID_LINUX): el motor, el almacén de columnas, ambos orígenes, la exportación, el JSON del diseño, el renderizador con la salida PNG y SVG sin interfaz, y el modelo de la lista de campos.
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, la salida PNG y SVG sin interfaz y el origen DataTable, en .NET Framework 4.7.2 y .NET 8 y 9, véase Descripción general de .NET.

Paquetes

sgcPivotGrid incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcPivotGridD13.dpk para Delphi 13 y sgcPivotGridC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcPivotGrid	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, dbrtl, sgcUICore
sgcPivotGridVCL	Runtime, VCL	sgcPivotGrid_VCL y sgcPivotGrid_FieldList_VCL (TsgcVCLPivotGrid, TsgcVCLPivotFieldList)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcPivotGrid
sgcPivotGridFMX	Runtime, FireMonkey, desde XE6	sgcPivotGrid_FMX y sgcPivotGrid_FieldList_FMX (TsgcFMXPivotGrid, TsgcFMXPivotFieldList)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcPivotGrid
dclsgcPivotGrid	Tiempo de diseño, VCL	Registra TsgcPivotSource_DataSet, TsgcPivotSource_Table, TsgcVCLPivotGrid y TsgcVCLPivotFieldList en la página de paleta **SGC PivotGrid**, con el editor de componentes, la lista desplegable de FieldName y los editores de colecciones.	designide, rtl, vcl, sgcPivotGrid, sgcPivotGridVCL

dclsgcPivotGridFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXPivotGrid y TsgcFMXPivotFieldList en la página de paleta **SGC PivotGrid**, con su editor de componentes.	designide, rtl, fmx, sgcPivotGrid, sgcPivotGridFMX, dclsgcPivotGrid
sgcPivotGridWS, dclsgcPivotGridWS	Runtime y tiempo de diseño, puente de sgcWebSockets	El puente de datos en vivo, independiente del framework.	rtl, sgcPivotGrid, sgcWebSockets

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcPivotGridFMX requiere dclsgcPivotGrid, así que instale primero dclsgcPivotGrid.

Versiones del IDE y plataformas de destino

Delphi	sgcPivotGrid	sgcPivotGridVCL	sgcPivotGridFMX	dclsgcPivotGrid, dclsgcPivotGridFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcPivotGrid, sgcPivotGridVCL	sgcPivotGridFMX	dclsgcPivotGrid, dclsgcPivotGridFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcPivotGrid)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcPivotGrid)
XE6 a 12	Win32, Win64	Win32, Win64	Win32

13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcPivotGridC13.cbproj, sgcPivotGridVCLC13.cbproj, sgcPivotGridFMXC13.cbproj, dclsgcPivotGridC13.cbproj y dclsgcPivotGridFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXPivotGrid y TsgcFMXPivotFieldList) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcPivotGridVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcPivotGrid_FMX y sgcPivotGrid_FieldList_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcPivotGrid se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa el motor pivot y las fuentes. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Restricciones de versión

- Todas las unidades compilan en Delphi 7: sin genéricos, sin métodos anónimos, sin variables en línea y sin ayudantes de registro.
- Los encabezados de C++Builder conservan prefijos de enumeración únicos (pva, pvf, pvd, pvx...), de modo que nunca chocan con sgcGrid ni con sgcCharts.

Véase también

- Ediciones
- Arquitectura

Inicio rápido VCL

Instale los paquetes de sgcPivotGrid para su versión de Delphi. Los componentes están en la página de paleta **SGC PivotGrid**.

En tiempo de diseño

1. Suelte un **TsgcVCLPivotGrid** en un formulario y establezca Align a alClient.
2. Suelte un **TsgcVCLPivotFieldList**, establezca Align a alRight y establezca su propiedad PivotGrid a la cuadrícula.
3. Suelte un **TsgcPivotSource_DataSet**, establezca DataSet a su conjunto de datos, PivotGrid a la cuadrícula y Active a True.
4. Expanda la propiedad Pivot de la cuadrícula y añada elementos a Pivot.Fields. Elija una columna en la lista desplegable de FieldName y establezca el Area: pvaRow, pvaColumn, pvaFilter o pvaData. Un campo de datos también tiene una Aggregation, Sum por defecto.
5. Ejecute la aplicación. El usuario reorganiza los campos con la lista de campos, y la cuadrícula recalcula después de cada cambio.

En código

La cuadrícula posee su motor, la propiedad Pivot. Cargue las filas entre BeginLoad y EndLoad, y cambie el diseño entre BeginUpdate y EndUpdate, de modo que cada bloque recalcule una sola vez.

```
uses
  SysUtils, Controls, Forms,
  sgcPivotGrid_Types, sgcPivotGrid_Engine, sgcPivotGrid_VCL,
  sgcPivotGrid_FieldList_VCL;

procedure CreatePivot(aForm: TForm);
var
  oGrid: TsgcVCLPivotGrid;
  oList: TsgcVCLPivotFieldList;
begin
  // the field list on the right, the grid fills the rest of the form
  oList := TsgcVCLPivotFieldList.Create(aForm);
  oList.Parent := aForm;
  oList.Align := alRight;
  oList.Width := 260;

  oGrid := TsgcVCLPivotGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := alClient;
  oList.PivotGrid := oGrid;

  // the source rows, one recalculation at EndLoad
  oGrid.Pivot.BeginLoad;
  try
    oGrid.Pivot.DefineColumn('Region', pvtText);
    oGrid.Pivot.DefineColumn('Product', pvtText);
    oGrid.Pivot.DefineColumn('Date', pvtDateTime);
    oGrid.Pivot.DefineColumn('Sales', pvtFloat);
    oGrid.Pivot.AppendRow(['North', 'Bikes', EncodeDate(2025, 1, 14), 1200.0]);
    oGrid.Pivot.AppendRow(['North', 'Helmets', EncodeDate(2025, 2, 3), 310.5]);
    oGrid.Pivot.AppendRow(['South', 'Bikes', EncodeDate(2025, 4, 22), 980.0]);
    oGrid.Pivot.AppendRow(['South', 'Helmets', EncodeDate(2026, 1, 9), 145.0]);
  finally
    oGrid.Pivot.EndLoad;
  end;

  // the layout, one recalculation at EndUpdate
```

```
  oGrid.Pivot.BeginUpdate;
  try
    oGrid.Pivot.AddField('Region', pvaRow);
    oGrid.Pivot.AddField('Product', pvaRow);
    oGrid.Pivot.AddField('Date', pvaColumn, pvdYear);
    oGrid.Pivot.AddDataField('Sales', pvfSum).DisplayFormat := '#,##0.00';
  finally
    oGrid.Pivot.EndUpdate;
  end;
end;
```

Véase también

- TsgcVCLPivotGrid
- TsgcVCLPivotFieldList
- Inicio rápido DataSet
- Áreas

Inicio rápido FMX

Los componentes de FireMonkey requieren Delphi XE6 o posterior. Están en la página de paleta **SGC PivotGrid** de un formulario de FireMonkey.

En tiempo de diseño

1. Suelte un **TsgcFMXPivotGrid** en un formulario y establezca Align a Client.
2. Suelte un **TsgcFMXPivotFieldList**, establezca Align a Right y establezca su propiedad PivotGrid a la cuadrícula.
3. Enlace los datos con un **TsgcPivotSource_DataSet** o un **TsgcPivotSource_Table**, o cárguelos en código, y añada los campos a Pivot.Fields como en Inicio rápido VCL.

La cuadrícula FMX dibuja sus propias barras de desplazamiento, que se muestran mientras se arrastran, y se desplaza con la rueda del ratón y con el gesto táctil. Su método de prueba de impacto es HitAt, porque FireMonkey ya publica una propiedad HitTest.

En código

```
uses
  SysUtils, FMX.Types, FMX.Controls, FMX.Forms,
  sgcUICore_Theme, sgcPivotGrid_Types, sgcPivotGrid_Engine, sgcPivotGrid_FMX,
  sgcPivotGrid_FieldList_FMX;

procedure CreatePivot(aForm: TForm);
var
  oGrid: TsgcFMXPivotGrid;
  oList: TsgcFMXPivotFieldList;
begin
  oList := TsgcFMXPivotFieldList.Create(aForm);
  oList.Parent := aForm;
  oList.Align := TAlignLayout.Right;
  oList.Width := 260;

  oGrid := TsgcFMXPivotGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := TAlignLayout.Client;
  oGrid.Theme := utDark;
  oList.PivotGrid := oGrid;
  oList.Theme := utDark;

  oGrid.Pivot.BeginLoad;
  try
    oGrid.Pivot.DefineColumn('Region', pvtText);
    oGrid.Pivot.DefineColumn('Channel', pvtText);
    oGrid.Pivot.DefineColumn('Date', pvtDateTime);
    oGrid.Pivot.DefineColumn('Sales', pvtFloat);
    oGrid.Pivot.AppendRow(['North', 'Online', EncodeDate(2025, 3, 2), 420.0]);
    oGrid.Pivot.AppendRow(['North', 'Retail', EncodeDate(2025, 7, 18), 610.0]);
    oGrid.Pivot.AppendRow(['South', 'Online', EncodeDate(2025, 11, 5), 275.0]);
  finally
    oGrid.Pivot.EndLoad;
  end;

  oGrid.Pivot.BeginUpdate;
  try
    oGrid.Pivot.AddField('Region', pvaRow);
    oGrid.Pivot.AddField('Date', pvaColumn, pvdQuarter);
    oGrid.Pivot.AddField('Channel', pvaFilter);
    oGrid.Pivot.AddDataField('Sales', pvfSum);
  finally
```

```
    oGrid.Pivot.EndUpdate;
  end;
end;
```

Véase también

- TsgcFMXPivotGrid
- TsgcFMXPivotFieldList
- Plataformas

Inicio rápido C++Builder

Instale los paquetes de sgcPivotGrid para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de inclusión y de biblioteca del proyecto, y use los componentes de la página de paleta **SGC PivotGrid** exactamente como en Delphi. El instalador añade ambas rutas al IDE por usted. Los encabezados .hpp y los archivos .lib generados de sgcPivotGrid y sgcUICore están en esa única carpeta, por lo que sgcUICore no necesita una ruta propia.

Cosas que conviene saber en C++

- AppendRow recibe un arreglo abierto de Variant, que C++ pasa como un arreglo de Variant más su índice más alto: `AppendRow(Row, 3)` para cuatro valores.
- El encabezado generado no tiene argumentos por defecto para los constructores de TsgcPivotEngine y TsgcPivotGridRenderer. Pase NULL de forma explícita: `new TsgcPivotEngine(NULL)` y `new TsgcPivotGridRenderer(NULL, NULL)`. Consulte Inicio rápido sin interfaz.
- Los eventos son closures, por lo que sus manejadores son funciones miembro __fastcall de una clase. Un parámetro var como aHandled es una referencia bool.

En código

Véase también

- TsgcVCLPivotGrid
- Inicio rápido VCL
- Demos

Inicio rápido DataSet

TsgcPivotSource_DataSet copia los registros de cualquier TDataSet en el motor de una cuadrícula y mantiene actualizada la tabla dinámica. El diseño se refiere a los campos del conjunto de datos por su FieldName, de modo que puede definirse antes de que lleguen los datos.

```
uses
  SysUtils, Classes, Controls, Forms, DB,
  sgcPivotGrid_Types, sgcPivotGrid_VCL, sgcPivotGrid_Source_DataSet;

procedure BindDataSet(aForm: TForm; aDataSet: TDataSet);
var
  oGrid: TsgcVCLPivotGrid;
  oSource: TsgcPivotSource_DataSet;
begin
  oGrid := TsgcVCLPivotGrid.Create(aForm);
  oGrid.Parent := aForm;
  oGrid.Align := alClient;

  // the layout names the fields of the dataset
  oGrid.Pivot.BeginUpdate;
  try
    oGrid.Pivot.AddField('Country', pvaRow);
    oGrid.Pivot.AddField('OrderDate', pvaColumn, pvdYear);
    oGrid.Pivot.AddField('OrderDate', pvaColumn, pvdQuarter);
    oGrid.Pivot.AddDataField('Amount', pvfSum);
    oGrid.Pivot.AddDataField('OrderNo', pvfCount);
  finally
    oGrid.Pivot.EndUpdate;
  end;

  // the source copies the dataset into the engine of the grid and
  // reloads it when the data changes
  oSource := TsgcPivotSource_DataSet.Create(aForm);
  oSource.Columns.CommaText := 'Country,OrderDate,Amount,OrderNo';
  oSource.DataSet := aDataSet;
  oSource.PivotGrid := oGrid;
  oSource.LiveUpdates := True;
  oSource.Active := True;
end;
```

Cómo carga el origen

- Columns enumera los campos del conjunto de datos que se cargan. Vacío carga todos los campos. Los campos BLOB, binarios y anidados siempre se omiten.
- Los tipos de columna proceden de los tipos de campo: los campos de cadena son texto, los campos enteros y de coma flotante son números, los campos de fecha y hora son fechas, los campos booleanos son booleanos.
- La carga se ejecuta con DisableControls y restaura el marcador, de modo que los controles enlazados al conjunto de datos no parpadean.
- Con LiveUpdates, un post, un append, un delete, un refresh, First o Last recarga la tabla dinámica, ya que TDataSet los notifica como cambios de datos. El desplazamiento con Next, Prior o MoveBy no recarga. Una carga se ejecuta solo mientras el conjunto de datos está en estado de exploración.
- Load recarga en cualquier momento, Refresh recarga cuando el origen está Active.

Un proyecto de C++Builder que usa TClientDataSet enlaza midas.lib con `#pragma link "midas.lib"`, como hace la demo DataSetPivot.

Véase también

- TsgcPivotSource_DataSet

- TsgcPivotSource_Table
- Agrupación de fechas

Inicio rápido sin interfaz

Una tabla dinámica no necesita una ventana. TsgcPivotEngine carga las filas y calcula, sus métodos de exportación escriben CSV, HTML y JSON, y TsgcPivotGridRenderer dibuja la misma cuadrícula que los controles en un PNG o un SVG. Estas unidades usan solo la RTL y sgcUICore, de modo que un informe se ejecuta en un servicio, en un servidor web o en Linux64.

Delphi

```
uses
  SysUtils, Classes,
  sgcUICore_Theme, sgcPivotGrid_Types, sgcPivotGrid_Engine,
  sgcPivotGrid_Render;

procedure WriteReport(const aFolder: string);
var
  oEngine: TsgcPivotEngine;
  oRenderer: TsgcPivotGridRenderer;
  vPath: string;
begin
  vPath := IncludeTrailingPathDelimiter(aFolder);
  oEngine := TsgcPivotEngine.Create;
  try
    oEngine.BeginLoad;
    try
      oEngine.DefineColumn('Region', pvtText);
      oEngine.DefineColumn('Channel', pvtText);
      oEngine.DefineColumn('Sales', pvtFloat);
      oEngine.AppendRow(['North', 'Online', 1200.0]);
      oEngine.AppendRow(['North', 'Retail', 800.0]);
      oEngine.AppendRow(['South', 'Online', 450.0]);
      oEngine.AppendRow(['South', 'Partner', 300.0]);
    finally
      oEngine.EndLoad;
    end;

    oEngine.BeginUpdate;
    try
      oEngine.AddField('Region', pvaRow);
      oEngine.AddField('Channel', pvaColumn);
      oEngine.AddDataField('Sales', pvfSum);
    finally
      oEngine.EndUpdate;
    end;

    // the whole view with its totals as tables
    oEngine.SaveToCSV(vPath + 'sales.csv');
    oEngine.SaveToHTML(vPath + 'sales.html');
    oEngine.SaveToJSON(vPath + 'sales.json');

    // the grid as an image, drawn on the sgcUICore canvases
    oRenderer := TsgcPivotGridRenderer.Create;
    try
      oRenderer.Options.Theme := utDark;
      oRenderer.SaveToPNG(oEngine, sgcPivotRenderState, vPath + 'sales.png');
      oRenderer.SaveToSVG(oEngine, sgcPivotRenderState, vPath + 'sales.svg');
    finally
      oRenderer.Free;
    end;
  finally
    oEngine.Free;
```

```
  end;
end;
```

Notas

- sgcPivotRenderState devuelve un estado de vista vacío: sin desplazamiento, sin selección y sin anulaciones de ancho de columna. Las sobrecargas del renderizador que reciben el motor lo bloquean mientras dibujan.
- Un ancho y un alto de 0 dibujan todo el contenido. Los tamaños están en DIP y el argumento de escala los multiplica.

Véase también

- Exportación
- TsgcPivotGridRenderer
- Demos

Demos

Las demos se instalan en `Demos\PivotGrid`, con los proyectos de Delphi en `Delphi` y sus ports de C++Builder en `CBuilder`. Todas las demos aceptan `--test`, que ejecuta su autoprueba y termina, y `/snapshot <file.png>`, que guarda una imagen de la tabla dinámica.

Demo	Framework	Qué muestra
SalesPivot	VCL	100.000 filas de ventas: agrupación de fechas por año, trimestre y mes, expandir y contraer, una ordenación por valor, una escala de color, la lista de campos y la exportación a CSV, HTML y PNG.
DataSetPivot	VCL	Una tabla dinámica sobre un TClientDataSet: una edición, un append o un delete en el TDBGrid la recarga, y un doble clic en una celda hace drill-down hasta los pedidos de origen.
MillionRows	VCL	1.000.000 de filas añadidas en un solo bloque de carga, CountDistinct de una columna con 100.000 valores y StdDev junto a Sum, con el recálculo cronometrado.
AdvancedPivot	VCL	Un campo calculado Profit, mostrar valores como, Top N con Others, y el diseño guardado y cargado desde un archivo JSON.
GridToPivotToChart	VCL	Los pedidos escritos en un sgcGrid alimentan la tabla dinámica mediante TsgcPivotSource_Table, y el resultado de la tabla dinámica alimenta sgcCharts mediante IsgcUITable.
SalesPivotFMX	FireMonkey	50.000 filas con el diseño SalesPivot, desplazamiento táctil, el tema oscuro y exportación a PNG.
HeadlessReport	Consola	Lee un archivo CSV y escribe report.png, report.svg, report.html, report.csv y report.json sin ningún framework de interfaz.
AskYourPivot	VCL	Una pregunta en lenguaje natural se convierte en el diseño del pivote mediante TsgcPivotGridAIQuery y un TsgcAIChat: los cuadros de proveedor, modelo y clave de API, Ask, Undo, preguntas de ejemplo y Show prompt. Edición Registered, requiere sgcWebSockets. Consulte Pregunte a su tabla dinámica.

Véase también

- Inicio rápido VCL
- Inicio rápido sin interfaz
- Pregunte a su tabla dinámica

TsgcVCLPivotGrid

Unidad: sgcPivotGrid_VCL
TsgcVCLPivotGrid es la tabla dinámica VCL, solo para Windows. Posee un TsgcPivotEngine, publicado como Pivot, y dibuja su resultado con filas y columnas virtuales: una esquina fija, encabezados de columna de varios niveles con combinación de celdas, encabezados de fila, botones + y -, totales y totales generales. Se desplaza con barras nativas y con la rueda del ratón (Mayús desplaza en horizontal), cambia el ancho de una columna por el borde de su encabezado y tiene una interfaz de teclado: las flechas, RePág, AvPág, Inicio y Fin mueven la selección, Derecha e Izquierda (o + y -) expanden y contraen un miembro cuando el encabezado de fila tiene el foco, y Enter hace drill-down. Hereda de TsgcUIVCLControl y publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop y Visible.

Propiedades

Propiedad	Tipo	Descripción
Pivot	TsgcPivotEngine	El motor que posee la cuadrícula: sus campos, datos, totales y títulos, guardados con el formulario. La cuadrícula implementa IsgcUITable delegando en él.
BorderStyle	TBorderStyle	bsSingle (por defecto) o bsNone.
RowLayout	TsgcPivotRowLayout	pvlCompact (por defecto) dibuja los niveles de fila en una columna de encabezado con sangría, pvlTabular da a cada nivel su propia columna. Consulte Áreas.
RowHeight	Integer	La altura de una fila en DIP. Por defecto 24.
HeaderHeight	Integer	La altura de una fila de encabezado de columna en DIP. Por defecto 26.
IndentWidth	Integer	La sangría de cada nivel de fila en el diseño compacto, en DIP. Por defecto 16.
MinColumnWidth, MaxColumnWidth	Integer	Los límites de los anchos automáticos de las columnas de datos, en DIP. Por defecto 48 y 320.
AutoColumnWidths	Boolean	Ajusta las columnas de datos a su contenido. Si es False miden 96 DIP de ancho. Las columnas de encabezado de fila siempre se ajustan. Por defecto True.
GridLines	Boolean	Dibuja las líneas de las celdas. Por defecto True.
AlternateRows	Boolean	Sombrea una fila de cada dos. Por defecto False.
TotalsBold	Boolean	Dibuja los totales en negrita. Por defecto True.
ShowExpandButtons	Boolean	Muestra los botones + y - de los miembros que tienen hijos. Por defecto True.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom.
AccentColor	TsgcUIColor	Un color de acento $AARRGGBB. 0 (por defecto) usa el acento del tema.

FontSize	Double	El tamaño de fuente de las celdas en puntos. Por defecto 12.
FormatRules	TsgcPivotFormatRules	Las reglas de formato condicional de las celdas.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
SelectedRow	Integer	Pública. El elemento de fila seleccionado, -1 si no hay ninguno.
SelectedCol	Integer	Pública. La columna de datos seleccionada, -1 cuando el encabezado de fila tiene el foco o no hay nada seleccionado.
TopRow	Integer	Pública. El primer elemento de fila mostrado bajo los encabezados de columna.
ScrollX	Integer	Pública. El desplazamiento horizontal de las columnas de datos.
Renderer	TsgcPivotGridRenderer	Pública, solo lectura. El renderizador que dibuja la cuadrícula.

Métodos

Método	Devuelve	Descripción
HitTest(X, Y)	TsgcPivotHitInfo	La parte bajo un punto en píxeles del dispositivo: Part (pvhNone, pvhCorner, pvhColumnHeader, pvhRowHeader, pvhCell, pvhExpandButton, pvhColumnResize), RowItem, DataCol y Level.
ScrollToCell(aRowItem, aDataCol)		Se desplaza hasta que la celda es visible. -1 deja ese eje como está.
Recalculate		Recalcula el motor ahora.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0)		Toda la vista como imagen, tamaños en DIP, 0 es el tamaño del contenido, hasta 16384 píxeles.
SaveToCSV, SaveToJSON, SaveToHTML(aFileName or aStream)		La vista actual como tabla. Consulte Exportación.
SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile, LoadLayoutFromFile		El diseño de Pivot como JSON, con los anchos de columna de la cuadrícula. Edición Advanced. Consulte Persistencia del diseño.

Eventos

Evento	Tipo	Descripción
OnCellClick, OnCellDblClick	TsgcVCLPivotHitEvent	Un clic o un doble clic, con la información de impacto del punto.

OnDrillDown	TsgcVCLPivot-DrillDownEvent	Enter o un doble clic en una celda: aRowItem, aDataCol y var aHandled. La cuadrícula no tiene acción integrada, llame a Pivot.DrillDown o Pivot.GetSourceRows para leer las filas de origen.
OnExpandChan-ge	TsgcVCLPivo-tExpandEvent	Después de que el usuario expandiera o contrajera un miembro: aA-xis y aItem.
OnGetCellStyle	TsgcPivot-CellStyleEvent	Antes de dibujar una celda de datos: aRowItem, aDataCol y var aBack, aFore y aBold, después de las reglas de formato.
OnGetCellText	TsgcPivotCell-TextEvent	Antes de dibujar una celda de datos: aRowItem, aDataCol y var aText, el valor formateado, o Pivot.EmptyText cuando la celda está vacía.
OnSelection-Change	TNotifyEvent	SelectedRow o SelectedCol han cambiado.
OnColumnResi-zed	TsgcVCLPivot-HitEvent	El usuario cambió el ancho de una columna arrastrando el borde de su encabezado.

Mientras se carga el formulario, el motor solo se marca como sucio, y la cuadrícula recalcula una vez en Loaded cuando Pivot.AutoRecalculate es True.

Ejemplo

```
uses
  SysUtils, Dialogs,
  sgcPivotGrid_Engine, sgcPivotGrid_VCL;

type
  TPivotEvents = class
  public
    procedure DrillDown(Sender: TObject; aRowItem, aDataCol: Integer;
      var aHandled: Boolean);
    procedure GetCellText(Sender: TObject; aRowItem, aDataCol: Integer;
      var aText: string);
  end;

procedure TPivotEvents.DrillDown(Sender: TObject; aRowItem, aDataCol: Integer;
  var aHandled: Boolean);
var
  oView: TsgcPivotSourceView;
begin
  // the source rows behind the cell, the caller frees the view
  oView := TsgcVCLPivotGrid(Sender).Pivot.DrillDown(aRowItem, aDataCol);
  try
    ShowMessage(Format('%d source rows', [oView.TableRowCount]));
  finally
    oView.Free;
  end;
  aHandled := True;
end;

procedure TPivotEvents.GetCellText(Sender: TObject; aRowItem, aDataCol: Integer;
  var aText: string);
begin
  // negative values in parentheses; an empty cell arrives as EmptyText
  if (Length(aText) > 1) and (aText[1] = '-') then
    aText := '(' + Copy(aText, 2, MaxInt) + ')';
end;

procedure HookEvents(aGrid: TsgcVCLPivotGrid; aEvents: TPivotEvents);
begin
```

```
  // empty cells show a dash
  aGrid.Pivot.EmptyText := '-';
  aGrid.OnDrillDown := aEvents.DrillDown;
  aGrid.OnGetCellText := aEvents.GetCellText;
end;
```

Véase también

- TsgcFMXPivotGrid
- TsgcVCLPivotFieldList
- Inicio rápido VCL

TsgcFMXPivotGrid

Unidad: sgcPivotGrid_FMX
TsgcFMXPivotGrid es la tabla dinámica de FireMonkey, desde Delphi XE6 (SGC_PIVOTGRID_FMX). Tiene el mismo motor, diseño, propiedades y eventos que TsgcVCLPivotGrid. Las diferencias proceden de FireMonkey: dibuja sus propias barras de desplazamiento, que se muestran mientras se arrastran, se desplaza con la rueda del ratón y con el gesto táctil, y su método de prueba de impacto es HitAt, porque FireMonkey ya publica una propiedad HitTest. Sus tipos de evento llevan el prefijo TsgcFMXPivot. Hereda de TsgcUIFMXControl y publica las propiedades y eventos de diseño habituales de FireMonkey.

Propiedades

Propiedad	Tipo	Descripción
Pivot	TsgcPivotEngine	El motor que posee la cuadrícula: sus campos, datos, totales y títulos, guardados con el formulario. La cuadrícula implementa IsgcUITable delegando en él.
RowLayout	TsgcPivotRowLayout	pvlCompact (por defecto) dibuja los niveles de fila en una columna de encabezado con sangría, pvlTabular da a cada nivel su propia columna. Consulte Áreas.
RowHeight	Integer	La altura de una fila en DIP. Por defecto 24.
HeaderHeight	Integer	La altura de una fila de encabezado de columna en DIP. Por defecto 26.
IndentWidth	Integer	La sangría de cada nivel de fila en el diseño compacto, en DIP. Por defecto 16.
MinColumnWidth, MaxColumnWidth	Integer	Los límites de los anchos automáticos de las columnas de datos, en DIP. Por defecto 48 y 320.
AutoColumnWidths	Boolean	Ajusta las columnas de datos a su contenido. Si es False miden 96 DIP de ancho. Las columnas de encabezado de fila siempre se ajustan. Por defecto True.
GridLines	Boolean	Dibuja las líneas de las celdas. Por defecto True.
AlternateRows	Boolean	Sombrea una fila de cada dos. Por defecto False.
TotalsBold	Boolean	Dibuja los totales en negrita. Por defecto True.
ShowExpandButtons	Boolean	Muestra los botones + y - de los miembros que tienen hijos. Por defecto True.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom.
AccentColor	TsgcUIColor	Un color de acento $AARRGGBB. 0 (por defecto) usa el acento del tema.
FontSize	Double	El tamaño de fuente de las celdas en puntos. Por defecto 12.
FormatRules	TsgcPivotFormatRules	Las reglas de formato condicional de las celdas.

MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
SelectedRow	Integer	Pública. El elemento de fila seleccionado, -1 si no hay ninguno.
SelectedCol	Integer	Pública. La columna de datos seleccionada, -1 cuando el encabezado de fila tiene el foco o no hay nada seleccionado.
TopRow	Integer	Pública. El primer elemento de fila mostrado bajo los encabezados de columna.
ScrollX	Integer	Pública. El desplazamiento horizontal de las columnas de datos.
Renderer	TsgcPivotGridRenderer	Pública, solo lectura. El renderizador que dibuja la cuadrícula.
VertBarVisible, HorzBarVisible	Boolean	Pública, solo lectura. True cuando el contenido necesita esa barra de desplazamiento.

Métodos

Método	Devuelve	Descripción
HitAt(X, Y)	TsgcPivotHitInfo	La parte bajo un punto en coordenadas locales: Part (pvhNone, pvhCorner, pvhColumnHeader, pvhRowHeader, pvhCell, pvhExpandButton, pvhColumnResize), RowItem, DataCol y Level.
ScrollToCell(aRowItem, aDataCol)		Se desplaza hasta que la celda es visible. -1 deja ese eje como está.
Recalculate		Recalcula el motor ahora.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0)		Toda la vista como imagen, tamaños en DIP, 0 es el tamaño del contenido, hasta 16384 píxeles.
SaveToCSV, SaveToJSON, SaveToHTML(aFileName or aStream)		La vista actual como tabla. Consulte Exportación.
SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile, LoadLayoutFromFile		El diseño de Pivot como JSON, con los anchos de columna de la cuadrícula. Edición Advanced. Consulte Persistencia del diseño.

Eventos

Evento	Tipo	Descripción
OnCellClick, OnCellDblClick	TsgcFMXPivotHitEvent	Un clic o un doble clic, con la información de impacto del punto.
OnDrillDown	TsgcFMXPivotDrillDownEvent	Enter o un doble clic en una celda: aRowItem, aDataCol y var aHandled. La cuadrícula no tiene acción integrada, llame a Pivot.DrillDown o Pivot.GetSourceRows para leer las filas de origen.

OnExpandChange	TsgcFMXPivotExpandEvent	Después de que el usuario expandiera o contrajera un miembro: aAxis y aItem.
OnGetCellStyle	TsgcPivotCellStyleEvent	Antes de dibujar una celda de datos: aRowItem, aDataCol y var aBack, aFore y aBold, después de las reglas de formato.
OnGetCellText	TsgcPivotCellTextEvent	Antes de dibujar una celda de datos: aRowItem, aDataCol y var aText, el valor formateado, o Pivot.EmptyText cuando la celda está vacía.
OnSelectionChange	TNotifyEvent	SelectedRow o SelectedCol han cambiado.
OnColumnResized	TsgcFMXPivotHitEvent	El usuario cambió el ancho de una columna arrastrando el borde de su encabezado.

Véase también

- TsgcVCLPivotGrid
- TsgcFMXPivotFieldList
- Inicio rápido FMX

TsgcVCLPivotFieldList

Unidad: sgcPivotGrid_FieldList_VCL
TsgcVCLPivotFieldList es la lista de campos VCL. Enlazada a una tabla dinámica, permite al usuario construir el diseño con arrastrar y soltar. Hereda de TsgcUIVCLControl y publica las propiedades de diseño habituales de VCL. Su lista de miembros es un TsgcVCLPivotMemberPopup y su menú de agregación un TPopupMenu.

Cómo trabaja el usuario con ella

- Las columnas de origen se enumeran arriba, con una marca para las columnas que están en el diseño. Debajo hay cuatro zonas: Filtros, Columnas, Filas y Valores.
- Arrastre una columna a una zona para añadir un campo. Arrastre una ficha a otra zona para moverla, o fuera de las zonas para quitarla.
- Una columna soltada en Valores recibe Count si es de texto y Sum en caso contrario. El botón de una ficha de Valores abre un menú con todas las agregaciones.
- El botón de filtro de una ficha de Filtros, Columnas o Filas abre la lista de miembros, con un cuadro de búsqueda y Select All. OK escribe FilterValues y FilterExclude, consulte Filtros.
- Los filtros de etiqueta y de valor se establecen en código o en el Object Inspector.

Propiedades

Propiedad	Tipo	Descripción
PivotGrid	TComponent	La tabla dinámica cuyo motor edita la lista, un TsgcVCLPivotGrid o un TsgcFMXPivotGrid.
Engine	TsgcPivotEngine	Pública. Un motor asignado en código tiene prioridad sobre el motor de PivotGrid, para un motor sin interfaz.
ShowFilterArea	Boolean	Muestra la zona de Filtros. Por defecto True.
DeferUpdate	Boolean	Reúne los cambios del usuario y los aplica solo en ApplyChanges, con un único recálculo. Por defecto False.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom.
AccentColor	TsgcUIColor	Un acento $AARRGGBB. 0 (por defecto) usa el acento del tema.
FontSize	Double	El tamaño de la fuente en puntos.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.

Métodos

Método	Devuelve	Descripción
ApplyChanges		Asigna el diseño pendiente al motor cuando DeferUpdate es True.

Eventos

Evento	Tipo	Descripción
OnBefore-FieldMove	TsgcPivotBe-foreFieldMo-veEvent	Antes de que el usuario mueva un campo: aFieldName, aFrom, aTo y var aAllow. Para una columna de origen soltada en una zona aFrom es igual a aTo. Quitar un campo no genera evento de movimiento.
OnField-Moved	TsgcPivot-FieldMovedE-vent	Después de añadir o mover un campo: aField.

Ejemplo

```
uses
  SysUtils, Controls, Forms,
  sgcPivotGrid_Types, sgcPivotGrid_VCL, sgcPivotGrid_FieldList_VCL;

type
  TFieldListRules = class
  public
    procedure BeforeFieldMove(Sender: TObject; const aFieldName: string;
      aFrom, aTo: TsgcPivotArea; var aAllow: Boolean);
  end;

procedure TFieldListRules.BeforeFieldMove(Sender: TObject;
  const aFieldName: string; aFrom, aTo: TsgcPivotArea; var aAllow: Boolean);
begin
  // the customer names never go to the columns
  if SameText(aFieldName, 'Customer') and (aTo = pvaColumn) then
    aAllow := False;
end;

function AddFieldList(aForm: TForm; aGrid: TsgcVCLPivotGrid;
  aRules: TFieldListRules): TsgcVCLPivotFieldList;
begin
  Result := TsgcVCLPivotFieldList.Create(aForm);
  Result.Parent := aForm;
  Result.Align := alRight;
  Result.Width := 260;
  Result.PivotGrid := aGrid;
  // collect the changes, the grid recalculates on ApplyChanges
  Result.DeferUpdate := True;
  Result.OnBeforeFieldMove := aRules.BeforeFieldMove;
end;
```

Véase también

- TsgcFMXPivotFieldList
- Áreas

TsgcFMXPivotFieldList

Unidad: sgcPivotGrid_FieldList_FMX
TsgcFMXPivotFieldList es la lista de campos de FireMonkey, desde Delphi XE6 (SGC_PIVOTGRID_FMX), con las mismas zonas, el mismo arrastrar y soltar y las mismas reglas de filtro que TsgcVCLPivotFieldList. Sus menús emergentes se abren cuando se suelta el botón sobre la misma ficha.

Cómo trabaja el usuario con ella

- Las columnas de origen se enumeran arriba, con una marca para las columnas que están en el diseño. Debajo hay cuatro zonas: Filtros, Columnas, Filas y Valores.
- Arrastre una columna a una zona para añadir un campo. Arrastre una ficha a otra zona para moverla, o fuera de las zonas para quitarla.
- Una columna soltada en Valores recibe Count si es de texto y Sum en caso contrario. El botón de una ficha de Valores abre un menú con todas las agregaciones.
- El botón de filtro de una ficha de Filtros, Columnas o Filas abre la lista de miembros, con un cuadro de búsqueda y Select All. OK escribe FilterValues y FilterExclude, consulte Filtros.
- Los filtros de etiqueta y de valor se establecen en código o en el Object Inspector.

Propiedades

Propiedad	Tipo	Descripción
PivotGrid	TComponent	La tabla dinámica cuyo motor edita la lista, un TsgcVCLPivotGrid o un TsgcFMXPivotGrid.
Engine	TsgcPivotEngine	Pública. Un motor asignado en código tiene prioridad sobre el motor de PivotGrid, para un motor sin interfaz.
ShowFilterArea	Boolean	Muestra la zona de Filtros. Por defecto True.
DeferUpdate	Boolean	Reúne los cambios del usuario y los aplica solo en ApplyChanges, con un único recálculo. Por defecto False.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom.
AccentColor	TsgcUIColor	Un acento $AARRGGBB. 0 (por defecto) usa el acento del tema.
FontSize	Double	El tamaño de la fuente en puntos.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
Model	TsgcPivotFieldListModel	Pública, solo lectura. El modelo independiente del framework con las zonas, las fichas y el estado de arrastre.
AggregationMenu, FilterPopup, FilterSearch, FilterSelectAll, FilterList	controles de menú y menú emergente	Pública, solo lectura. El menú de agregación y las partes de la lista de miembros, creados la primera vez que se muestran.

Métodos

Método	Devuelve	Descripción
ApplyChanges		Asigna el diseño pendiente al motor cuando DeferUpdate es True, con un único recálculo.
HitAt(X, Y)	TsgcPivotFieldListHitInfo	La parte de la lista bajo un punto local.
ShowAggregationMenu(aField)		Abre el menú de agregación de un campo de Valores.
ShowMemberFilter(aField)		Abre la lista de miembros de un campo de Filtros, Columnas o Filas.
CloseMemberFilter(aApply)		Cierra la lista de miembros. aApply escribe los miembros marcados en el filtro.

Eventos

Evento	Tipo	Descripción
OnBeforeFieldMove	TsgcPivotBeforeFieldMoveEvent	Antes de que el usuario mueva un campo: aFieldName, aFrom, aTo y var aAllow. Para una columna de origen soltada en una zona aFrom es igual a aTo. Quitar un campo no genera evento de movimiento.
OnFieldMoved	TsgcPivotFieldMovedEvent	Después de añadir o mover un campo: aField.

Véase también

- TsgcVCLPivotFieldList
- TsgcFMXPivotGrid

TsgcPivotEngine

Unidad: sgcPivotGrid_Engine
TsgcPivotEngine guarda las filas de origen en un almacén de columnas tipado, el diseño y el resultado calculado. Las cuadrículas poseen uno, publicado como Pivot, y también funciona por sí solo, sin ninguna interfaz. Desciende de TPersistent, implementa IsgcUITable sin recuento de referencias y se crea con `Create(aOwner: TPersistent = nil)`. En C++ pase NULL de forma explícita.

Propiedades

Propiedad	Tipo	Descripción
Fields	TsgcPivot-Fields	Los campos del diseño, en las cuatro áreas.
Calculated-Fields	TsgcPivot-Calculated-Fields	Los campos calculados. Edición Advanced.
RowGrandTo-tals, Column-GrandTotals	Boolean	Muestran la fila y la columna del total general. Valor por defecto True.
SubtotalPosition	TsgcPi-votSubtotal-Position	pvpTop (por defecto) o pvpBottom. Consulte Totales.
AutoRecalcula-te	Boolean	Recalcula tras cada cambio de los datos o del diseño, una vez por blo-que. Si es False, llame a Recalculate. Valor por defecto True.
GrandTotalCap-tion	string	El título de los totales generales. Por defecto Grand Total.
BlankCaption	string	La etiqueta del miembro vacío. Por defecto (blank).
EmptyText	string	El texto de una celda de datos sin valor, vacío por defecto. Las cuadrí-culas, el renderizador, la exportación HTML y ResultTable lo muestran, y OnGetCellText lo recibe y puede sustituirlo. SaveToCSV escribe un campo vacío y SaveToJSON null. Un cambio vuelve a publicar la vista sin recalcular.
OthersCaption	string	La etiqueta del miembro Others de Top N. Por defecto Others. Edición Advanced.
MaxCells	Integer	El mayor número de celdas que puede crear un diseño. Un recálculo que necesite más genera EsgcPivotError. Por defecto 5.000.000.
Result	TsgcPivotRe-sult	Pública, solo lectura. La instantánea actual, nil antes del primer cálculo. Léala solo entre Lock y Unlock.
ResultTable	TsgcPivotRe-sultTable	Pública, solo lectura. La vista IsgcUITable del resultado. Consulte Ex-portación.
Store	TsgcPivotS-tore	Pública, solo lectura. El almacén de columnas, usado por los orígenes.

RowLayout	TsgcPivotRowLayout	Pública. La establecen las cuadrículas: en el diseño compacto un elemento padre va antes que sus hijos.

Métodos: datos

Método	Devuelve	Descripción
BeginLoad(aClear = True), EndLoad		Un bloque de carga. aClear vacía antes el almacén. El motor recalcula una sola vez, en el EndLoad más externo.
DefineColumn(aName), DefineColumn(aName, aType)	Integer	Añade una columna. Sin tipo, lo decide el primer valor que no esté vacío. Los nombres son únicos, sin distinguir mayúsculas.
AppendRow(aValues)	Integer	Añade una fila de valores Variant en el orden de las columnas. Null y los valores que no se pueden convertir quedan vacíos.
ClearData		Elimina todas las filas y columnas.
LoadFromTable(aTable)		Carga todas las columnas y filas de un IsgcUITable.
SourceRowCount, SourceColumnCount	Integer	El tamaño del almacén.
SourceColumnName(aIndex)	string	El nombre de una columna.
SourceValue(aRow, aCol)	Variant	Un valor almacenado, Null si está vacío.
GetMemberLabels(aField, aList)		Las etiquetas de miembro con las que compara un filtro de miembros de aField, en orden de visualización, con BlankCaption al final cuando un valor está vacío.

Métodos: diseño y vista

Método	Devuelve	Descripción
BeginUpdate, EndUpdate		Un bloque de diseño con un solo recálculo en EndUpdate.
Recalculate		Recalcula ahora.
AddField(aFieldName, aArea, aGrouping = pvdNone)	TsgcPivotField	Añade un campo de fila, columna, filtro o datos al final de su área.
AddDataField(aFieldName, aAggregation = pvfSum)	TsgcPivotField	Añade una medida.
MoveField(aField, aArea, aIndex)		Mueve un campo a otra área o posición.
RemoveField(aField), ClearLayout		Elimina un campo, o todos.
ExpandAll(aAxis), CollapseAll(aAxis)		Expande o contrae todos los miembros de pvxRows o pvxColumns.

ExpandLevel(aAxis, aLevel), CollapseLevel(aAxis, aLevel)		Expande o contrae todos los miembros de un nivel.
SetExpanded(aAxis, aItem, aValue), ToggleExpanded(aAxis, aItem)		Cambia un elemento de la instantánea actual. El estado expandido se conserva por ruta de miembro, de modo que sobrevive a un recálculo. Consulte Expandir y contraer.
GetExpandState(aAxis, aState), SetExpandState(aAxis, aState)		El estado expandido explícito de un eje como lista de cadenas. Edición Advanced.

Métodos: resultado, drill-down y exportación

Método	Devuelve	Descripción
Lock, Unlock		Protegen la instantánea mientras se lee Result.
GetSourceRows(aRowItem, aDataCol, var aRows)	Integer	Las filas de origen de una celda, en el orden de las filas. aDataCol -1 toma todas las columnas, y un total toma todas las filas que tiene debajo.
DrillDown(aRowItem, aDataCol)	TsgcPivotSourceView	Las mismas filas como un IsgcUITable con todas las columnas del almacén. El llamador lo libera, antes que el motor. No tiene filas después de que cambien los datos.
SaveToCSV, SaveToJSON, SaveToHTML		La vista actual, a archivo o a flujo. Consulte Exportación.
SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile, LoadLayoutFromFile		El diseño como JSON. Edición Advanced. Consulte Persistencia del diseño.
Assign(Source)		Copia el diseño y la configuración de otro motor, no sus datos.

Eventos

Evento	Tipo	Descripción
OnCalculated	TNotifyEvent	Después de publicar un nuevo resultado, en el hilo que recalculó.
OnLayoutChange	TNotifyEvent	Después de cambiar los campos o la configuración del diseño.
OnFilterRow	TsgcPivotFilterRowEvent	Para cada fila de origen mientras está asignado: aRow y var aAccept. Consulte Filtros.

Ejemplo

Lectura del resultado: las etiquetas de fila y los textos de las celdas, entre Lock y Unlock.

```
uses
  SysUtils, Classes,
  sgcPivotGrid_Types, sgcPivotGrid_Result, sgcPivotGrid_Engine;

procedure ListRows(aEngine: TsgcPivotEngine; aLines: TStrings);
```

```
var
  oResult: TsgcPivotResult;
  oItem: TsgcPivotItem;
  i, j: Integer;
  vLine: string;
begin
  aEngine.Lock;
  try
    oResult := aEngine.Result;
    if oResult = nil then
      Exit; // not calculated yet
    for i := 0 to oResult.ItemCount(pvxRows) - 1 do
    begin
      oItem := oResult.Item(pvxRows, i);
      if oItem.Kind = pvkGrandTotal then
        vLine := aEngine.GrandTotalCaption
      else
        vLine := StringOfChar(' ', oItem.Level * 2) +
          oResult.NodeLabel(pvxRows, oItem.Node);
      for j := 0 to oResult.DataColCount - 1 do
        vLine := vLine + #9 + oResult.CellText(i, j);
      aLines.Add(vLine);
    end;
  finally
    aEngine.Unlock;
  end;
end;
```

Véase también

- TsgcPivotField
- Inicio rápido sin interfaz
- Arquitectura

TsgcPivotField

Unidad: sgcPivotGrid_Fields
TsgcPivotField es un elemento de TsgcPivotEngine.Fields (TsgcPivotFields), un campo del diseño. El mismo FieldName puede aparecer en varios campos.

Propiedades

Propiedad	Tipo	Descripción
FieldName	string	La columna de origen o el campo calculado.
Caption	string	El título del nivel o de la medida. Si está vacío se usa DisplayCaption.
DisplayCaption	string	Pública, solo lectura. Caption, si no Sum of Sales para un campo de datos, si no FieldName.
Area	TsgcPivotArea	pvaRow (por defecto), pvaColumn, pvaData o pvaFilter. Consulte Áreas.
AreaIndex	Integer	La posición en el área. Para un campo de datos es el índice de la medida.
DateGrouping	TsgcPivotDateGrouping	pvdNone (por defecto), pvdYear, pvdQuarter, pvdMonth, pvdWeek o pvdDay. Consulte Agrupación de fechas.
DisplayFormat	string	Una imagen de formato FormatDateTime o FormatFloat para las etiquetas de miembro, o para los valores de un campo de datos.
Aggregation	TsgcPivotAggregation	La función de un campo de datos, pvfSum por defecto. Consulte Agregaciones.
SortBy	TsgcPivotSortBy	pvsLabel (por defecto) ordena los miembros por etiqueta, los números y las fechas por valor, pvsValue por su total.
SortOrder	TsgcPivotSortOrder	pvoAscending (por defecto) o pvoDescending.
SortMeasure	Integer	El campo de datos, por AreaIndex, de una ordenación por valor y de Top N. Por defecto 0.
Subtotals	Boolean	Muestra los subtotales de los miembros. Por defecto True.
Expanded	Boolean	El estado expandido inicial de los miembros de este nivel. Por defecto True.
FilterValues	TStrings	El filtro de miembros. Consulte Filtros.
FilterExclude	Boolean	FilterValues enumera los miembros que se excluyen. Por defecto False.
LabelFilter	TsgcPivotLabelFilter	Operator, Value y Value2 sobre las etiquetas de miembro.
ValueFilter	TsgcPivotValueFilter	Operator, Value, Value2 y Measure sobre los totales de miembro.

TopN, TopNOthers	Integer, Boolean	Top N con un miembro Others opcional. Edición Advanced. Consulte Top N.
ShowAs, ShowAsAxis, ShowAsBaseItem	TsgcPivotShowAs, TsgcPivotAxisKind, string	Mostrar valores como. Edición Advanced. Consulte Mostrar valores como.

Métodos

Método	Devuelve	Descripción
MoveTo(aArea, aAreaIndex)		Mueve el campo con una sola notificación de cambio. MaxInt es el final del área.
Assign(Source)		Copia todas las propiedades, incluidas Area y AreaIndex.

TsgcPivotFields

Método	Devuelve	Descripción
Add	TsgcPivotField	Añade un campo.
AreaCount(aArea)	Integer	El número de campos de un área.
FieldAt(aArea, aAreaIndex)	TsgcPivotField	El campo en una posición de un área.
Items[aIndex]	TsgcPivotField	Los campos en el orden de la colección.

Dentro de BeginUpdate y EndUpdate, y mientras se carga un formulario, los valores de AreaIndex se conservan tal como se dan. EndUpdate renumera cada área por AreaIndex y luego por el orden de la colección.

Véase también

- TsgcPivotEngine
- TsgcPivotCalculatedField

TsgcPivotCalculatedField

Unidad: sgcPivotGrid_Fields
TsgcPivotCalculatedField es un elemento de TsgcPivotEngine.CalculatedFields (TsgcPivotCalculatedFields): una columna Double calculada para cada fila de origen a partir de una expresión, que los campos usan mediante FieldName = Name como una columna de origen. Edición Advanced, consulte Campos calculados para la sintaxis.

Propiedades

Propiedad	Tipo	Descripción
Name	string	El nombre al que se refieren los campos. No debe ser el nombre de una columna de origen.
Expression	string	La expresión, por ejemplo [Revenue] - [Cost].

TsgcPivotCalculatedFields

Método	Devuelve	Descripción
Add	TsgcPivotCalculatedField	Añade un campo calculado.
IndexOfName(aName)	Integer	El índice de un campo calculado, -1 si no existe.
Items[aIndex]	TsgcPivotCalculatedField	Los campos calculados.

Véase también

- Campos calculados
- TsgcPivotField

TsgcPivotSource_DataSet

Unidad: sgcPivotGrid_Source_DataSet
TsgcPivotSource_DataSet es un componente no visual que copia un TDataSet en un motor de tabla dinámica mediante un TDataLink, y lo vuelve a cargar cuando cambian los datos. Consulte Inicio rápido DataSet.

Propiedades

Propiedad	Tipo	Descripción
DataSet	TDataSet	El conjunto de datos que se carga.
Active	Boolean	Carga los datos ahora y los mantiene actualizados. Por defecto False.
Columnas	TStrings	Las columnas que se cargan. Vacío carga todas las columnas.
LiveUpdates	Boolean	Recarga cuando cambian los datos mientras Active. Por defecto True.
PivotGrid	TCompo-nent	La tabla dinámica cuyo motor recibe los datos.
Engine	TsgcPivo-tEngine	Pública. El motor en el que escribe el origen: un motor asignado en código, si no el motor de PivotGrid.

Métodos

Método	Devuelve	Descripción
Load		Carga el conjunto de datos ahora, esté Active o no.
Refresh		Recarga cuando Active.
FieldTypeToDataType(aFieldType)	TsgcPivotDa-taType	Función de clase. El tipo de columna de un tipo de campo DB.

- La carga usa DisableControls y restaura el marcador. Los campos BLOB, binarios y anidados se omiten.
- Con LiveUpdates la tabla dinámica se recarga cuando el conjunto de datos notifica un cambio de datos en estado de exploración: un post, un append, un delete, un refresh, First o Last. Next, Prior y MoveBy no re-cargan.

Véase también

- TsgcPivotSource_Table
- Inicio rápido DataSet

TsgcPivotSource_Table

Unidad: sgcPivotGrid_Source_Table
TsgcPivotSource_Table copia cualquier IsgcUITable en un motor de tabla dinámica: un TsgcGridData, un TsgcVCLGrid o TsgcFMXGrid de sgcGrid, el resultado de otra tabla dinámica o su propia clase. Escucha la tabla y la recarga, bajo TableLock, cuando TableVersion cambió. La recarga se ejecuta en el hilo que notifica el cambio.

Propiedades

Propiedad	Tipo	Descripción
Table	TComponent	Un componente que implementa IsgcUITable.
Active	Boolean	Carga los datos ahora y los mantiene actualizados. Por defecto False.
Columnas	TStrings	Las columnas que se cargan. Vacío carga todas las columnas.
LiveUpdates	Boolean	Recarga cuando cambian los datos mientras Active. Por defecto True.
PivotGrid	TComponent	La tabla dinámica cuyo motor recibe los datos.
Engine	TsgcPivotEngine	Pública. El motor en el que escribe el origen: un motor asignado en código, si no el motor de PivotGrid.

Métodos

Método	Devuelve	Descripción
SetTable(aTable)		Enlaza cualquier objeto que implemente IsgcUITable. nil lo desenlaza.
GetTable	IsgcUITable	La tabla enlazada.
Load		Carga la tabla ahora, esté Active o no, incluso cuando su versión no cambió.
Refresh		Recarga cuando Active.

Las columnas con el mismo nombre que una columna anterior se omiten.

Ejemplo

```
uses
  Classes,
  sgcUICore_Table, sgcPivotGrid_VCL, sgcPivotGrid_Source_Table;

procedure BindTable(aPivot: TsgcVCLPivotGrid; aTable: TComponent);
var
  oSource: TsgcPivotSource_Table;
begin
  // aTable is any component that implements IsgcUITable,
  // for example a TsgcVCLGrid of sgcGrid
  oSource := TsgcPivotSource_Table.Create(aPivot.Owner);
  oSource.Table := aTable;
  oSource.PivotGrid := aPivot;
```

```
  oSource.Active := True;
end;

procedure BindTableObject(aPivot: TsgcVCLPivotGrid; const aTable: IsgcUITable;
  aSource: TsgcPivotSource_Table);
begin
  // an object that is not a component binds through SetTable
  aSource.SetTable(aTable);
  aSource.PivotGrid := aPivot;
  aSource.Active := True;
end;
```

Véase también

- TsgcPivotSource_DataSet
- Exportación

TsgcPivotGridRenderer

Unidad: sgcPivotGrid_Render
TsgcPivotGridRenderer dibuja el resultado de una tabla dinámica en cualquier lienzo de sgcUICore, y es el que usan las cuadrículas VCL y FMX. Por sí solo produce la salida PNG y SVG sin interfaz. Créelo con `Create(aOptions: TsgcPivotRenderOptions = nil; aFormatRulesOwner: TPersistent = nil)`. En C++ pase NULL en ambos. Consulte Inicio rápido sin interfaz.

Propiedades

Propiedad	Tipo	Descripción
Opciones	TsgcPivotRenderOptions	RowLayout, RowHeight, HeaderHeight, IndentWidth, MinColumnWidth, MaxColumnWidth, AutoColumnWidths, GridLines, AlternateRows, TotalsBold, ShowExpandButtons, Theme, AccentColor y FontSize, con el mismo significado y los mismos valores por defecto que en TsgcVCLPivotGrid.
FormatRules	TsgcPivotFormatRules	Las reglas de formato condicional.
GrandTotalCaption, TotalCaption	string	Los títulos de los totales generales y de los subtotales inferiores. Las sobrecargas del motor de SaveToPNG y SaveToSVG copian GrandTotalCaption del motor.
CurrentResult	TsgcPivotResult	Solo lectura. El resultado del último Render.

Métodos

Método	Devuelve	Descripción
Render(aResult, aState, aCanvas, aWidth, aHeight, aScale)		Dibuja la parte visible de un resultado, aWidth y aHeight en píxeles del dispositivo, aScale píxeles del dispositivo por DIP.
SaveToPNG, SaveToSVG(aResult or aEngine, aState, aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 1)		Todo el contenido como imagen, 0 es el tamaño del contenido. Las sobrecargas del motor bloquean el motor mientras dibujan.
HitTest(aResult, aState, aX, aY)	TsgcPivotHitInfo	La parte bajo un punto.
ContentWidth, ContentHeight	Integer	El tamaño de todo el contenido en DIP.
FrozenWidth, FrozenHeight	Integer	El tamaño de los encabezados de fila y de los encabezados de columna.
VisibleRange	Boolean	Los primeros y últimos elementos de fila y columnas de datos visibles de un área de visualización.

ColumnKey(aResult, aDataCol)	string	La clave de una columna de datos, usada por las anulaciones de ancho de columna.

Eventos

Evento	Tipo	Descripción
OnGet-CellStyle	TsgcPivotCellSty-leEvent	Los colores y el estado de negrita de una celda de datos, después de las reglas de formato.
OnGetCell-Text	TsgcPivotCellTex-tEvent	El texto de una celda de datos: el valor formateado, o el EmptyText del resultado cuando la celda está vacía.

Auxiliares

Rutina	Descripción
sgcPivotRenderState	Un TsgcPivotRenderState vacío: sin desplazamiento, selección, hover ni anulaciones de ancho de columna.
sgcPivotSetColumnWidth(var aState, aKey, aWidth)	Establece el ancho en DIP de la columna de datos cuyo ColumnKey es aKey.

Véase también

- Inicio rápido sin interfaz
- TsgcPivotFormatRule
- Exportación

TsgcPivotFormatRule

Unidad: sgcPivotGrid_Format
TsgcPivotFormatRule es un elemento de FormatRules (TsgcPivotFormatRules) de las cuadrículas y del renderizador: una regla de valor que colorea las celdas cuyo valor cumple una condición, o una escala de color entre dos colores.

Propiedades

Propiedad	Tipo	Descripción
Kind	TsgcPivotFormatRuleKind	pvrValue (por defecto), una regla de valor, o pvrColorScale.
Measure	Integer	El campo de datos, por AreaIndex, al que se aplica la regla. -1 (por defecto) se aplica a todas las medidas.
Operator, Value, Value2	TsgcPivotFilterOperator, Double, Double	La condición, con los operadores de los filtros. pvcNone coincide con todo valor que no esté vacío. Una escala de color también la usa para seleccionar sus valores.
BackColor, ForeColor	TsgcUIColor	Los colores $AARRGGBB de una celda coincidente. 0 (por defecto) conserva el color. Una escala de color solo usa ForeColor.
Bold	Boolean	Dibuja en negrita una celda coincidente. Por defecto False.
MinColor, MaxColor	TsgcUIColor	Los colores de una escala en el valor más pequeño y en el más grande de las celdas de miembro de la medida. Un MinColor transparente se desvanece hacia MaxColor.
Scope	TsgcPivotFormatScopes	Las celdas a las que se aplica la regla: pveCells (por defecto, celdas de miembro), pveTotals y pveGrandTotals.
Enabled	Boolean	Por defecto True.
StopOnMatch	Boolean	Una coincidencia termina la evaluación de las reglas para esa celda. Por defecto False.

Evaluación

- Las reglas se ejecutan en el orden de la colección. Cada regla activa que coincide aplica sus colores distintos de 0, de modo que una coincidencia posterior anula a una anterior, y Bold se acumula.
- Una celda vacía no coincide con ninguna regla y nunca se colorea.
- El rango de una escala de color se toma sobre las celdas de miembro, de modo que los totales fuera de él reciben los colores de los extremos.
- OnGetCellStyle se ejecuta después de las reglas y puede cambiar el resultado.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Format, sgcPivotGrid_VCL;

procedure AddRules(aGrid: TsgcVCLPivotGrid);
var
  oRule: TsgcPivotFormatRule;
```

```
begin
  // a colour scale over the member cells of the first measure
  oRule := aGrid.FormatRules.Add;
  oRule.Kind := pvrColorScale;
  oRule.Measure := 0;
  oRule.MinColor := $FFFFFFFF;
  oRule.MaxColor := $FF2E7D32;

  // negative values in red and bold, on the cells and the totals
  oRule := aGrid.FormatRules.Add;
  oRule.Kind := pvrValue;
  oRule.Operator := pvcLess;
  oRule.Value := 0;
  oRule.ForeColor := $FFC62828;
  oRule.Bold := True;
  oRule.Scope := [pveCells, pveTotals];
end;
```

Véase también

- TsgcVCLPivotGrid
- TsgcPivotGridRenderer

Áreas

Cada campo del diseño está en una de cuatro áreas. La propiedad Area de TsgcPivotField y el segundo argumento de AddField la eligen.

Area	Significado
pvaRow	Los miembros se convierten en los encabezados de fila. Varios campos de fila se anidan, el primero es el nivel más externo.
pvaColumn	Los miembros se convierten en los encabezados de columna, anidados del mismo modo.
pvaData	Una medida: los valores agregados en cada celda. Varios campos de datos dan varias columnas de medida bajo cada miembro de columna.
pvaFilter	Un filtro de informe. Sus miembros no se muestran, solo restringe las filas de origen mediante sus filtros de miembro y de etiqueta.

Orden y movimiento

- AreaIndex ordena los campos de un área. AddField, AddDataField y asignar Area colocan un campo al final de su nueva área.
- MoveField(aField, aArea, aIndex) mueve un campo a otra área o posición con una sola notificación. RemoveField lo elimina y ClearLayout quita todos los campos.
- Fields.AreaCount(aArea) y Fields.FieldAt(aArea, aIndex) leen el diseño área por área.
- El mismo FieldName puede aparecer en varios campos, por ejemplo una columna Date agrupada por año y por mes, o una columna Sales con Sum y Average.
- El índice de medida que usan SortMeasure, ValueFilter.Measure y las reglas de formato es el AreaIndex del campo de datos.

Nombres desconocidos

Un campo de fila, columna o filtro cuyo FieldName no es una columna de origen ni un campo calculado se ignora. Un campo de datos con un nombre desconocido conserva una medida vacía, de modo que los índices de medida no se desplazan. Esto permite definir un diseño antes de cargar los datos.

Filas compactas y tabulares

RowLayout de las cuadrículas decide cómo se dibujan los niveles de fila. pvlCompact (el valor por defecto) muestra cada nivel en una columna de encabezado de fila con sangría, con cada elemento padre en su propia fila. pvlTabular da a cada nivel de fila su propia columna de encabezado.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure ArrangeFields(aEngine: TsgcPivotEngine);
var
  oRegion, oYear: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    aEngine.ClearLayout;
```

```
    aEngine.AddField('Country', pvaFilter);
    oRegion := aEngine.AddField('Region', pvaRow);
    aEngine.AddField('Product', pvaRow);
    oYear := aEngine.AddField('Date', pvaColumn, pvdYear);
    aEngine.AddDataField('Sales', pvfSum);

    // swap the axes: the regions go to the columns, the years to the rows
    aEngine.MoveField(oRegion, pvaColumn, 0);
    aEngine.MoveField(oYear, pvaRow, 0);
  finally
    aEngine.EndUpdate;
  end;
end;
```

Cada cambio del diseño genera OnLayoutChange del motor. Dentro de BeginUpdate y EndUpdate el motor recalcula una sola vez, en EndUpdate.

Véase también

- Agregaciones
- TsgcPivotEngine
- TsgcVCLPivotFieldList

Agregaciones

Un campo de datos agrega su columna de origen con la función de Aggregation. Todas las funciones omiten los valores vacíos.

Aggregation	Result
pvfSum	La suma de los valores. El valor por defecto.
pvfCount	El número de valores que no están vacíos, como en Excel. Funciona con todos los tipos de columna.
pvfAverage	Sum dividido por Count.
pvfMin, pvfMax	El valor más pequeño y el más grande.
pvfCountDistinct	El número de valores distintos que no están vacíos. En una columna de texto cuenta etiquetas distintas.
pvfStdDev, pvfVar	La desviación estándar y la varianza muestrales (`n - 1`). Una celda con menos de dos valores está vacía.

Los totales son exactos

Los subtotales y los totales generales se calculan a partir de los valores de origen, no de las celdas que tienen debajo. El promedio de una región es el promedio de sus filas, no el promedio de los promedios de sus productos, y un recuento de distintos de un total cuenta cada valor una vez.

Títulos y formatos

- Sin Caption, un campo de datos se muestra como *<agregación> of <FieldName>*, por ejemplo Sum of Sales. DisplayCaption devuelve el título en uso.
- DisplayFormat es una imagen de formato FormatFloat para los valores de la medida, por ejemplo `#,##0.00`.
- Las medidas siempre están en las columnas. Con varios campos de datos, las columnas de medida son el nivel de columna más interno.

La lista de campos añade Count cuando el usuario suelta una columna de texto en Valores y Sum en caso contrario, y sus fichas de Valores abren un menú con todas las agregaciones.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure AddMeasures(aEngine: TsgcPivotEngine);
var
  oField: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    aEngine.AddField('Region', pvaRow);

    oField := aEngine.AddDataField('Sales', pvfSum);
    oField.DisplayFormat := '#,##0.00';

    // the same column a second time, with another aggregation
    oField := aEngine.AddDataField('Sales', pvfAverage);
```

```
    oField.Caption := 'Average ticket';
    oField.DisplayFormat := '0.00';

    aEngine.AddDataField('Customer', pvfCountDistinct);
    aEngine.AddDataField('OrderNo', pvfCount);
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- Mostrar valores como
- Totales
- TsgcPivotField

Agrupación de fechas

Un campo de fila, columna o filtro sobre una columna de fecha agrupa las fechas con DateGrouping, el tercer argumento de AddField.

DateGrouping	Etiqueta del miembro	Etiqueta bajo un nivel más grueso del mismo campo
pvdNone	Cada valor distinto de fecha y hora es su propio miembro.	
pvdYear	2024	
pvdQuarter	2024 Q1	Q1
pvdMonth	2024-01	Jan, el nombre corto del mes
pvdWeek	2024-W05, la semana ISO 8601 y su año	W05
pvdDay	La fecha corta	

- Cuando el nivel anterior del mismo eje es la misma columna con una agrupación más gruesa, la etiqueta omite el año, como muestra la última columna. Year, Quarter y Month de una misma columna Date dan 2024, Q1, Jan.
- Con un DisplayFormat, la etiqueta es el inicio del grupo formateado con FormatDateTime y esa imagen de formato.
- Los miembros de fecha se ordenan por fecha, no por etiqueta, de modo que 2024-10 va después de 2024-09.
- Una fila con fecha vacía pertenece al miembro en blanco, que se muestra con BlankCaption y se ordena al final.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Engine;

procedure GroupDates(aEngine: TsgcPivotEngine);
begin
  aEngine.BeginUpdate;
  try
    aEngine.AddField('Region', pvaRow);

    // the same Date column three times, from the coarsest grouping down;
    // the quarters start collapsed, so the months show on demand
    aEngine.AddField('Date', pvaColumn, pvdYear);
    aEngine.AddField('Date', pvaColumn, pvdQuarter).Expanded := False;
    aEngine.AddField('Date', pvaColumn, pvdMonth);

    aEngine.AddDataField('Sales', pvfSum);
  finally
    aEngine.EndUpdate;
  end;
end;
```

Expanded es el estado inicial de un nivel: aquí cada trimestre empieza contraído y el usuario abre los meses de un trimestre con su botón +.

Véase también

- Áreas
- Totales

Totales

Una tabla dinámica muestra un subtotal para cada miembro que tiene hijos y un total general para cada eje.

Propiedad	Efecto
TsgcPivotEngine.RowGrandTotals	Muestra la fila del total general. Por defecto True.
TsgcPivotEngine.ColumnGrandTotals	Muestra la columna del total general. Por defecto True.
TsgcPivotEngine.SubtotalPosition	pvpTop (por defecto): el subtotal está en el propio miembro, antes de sus hijos, y en un diseño compacto lo lleva la fila del padre. pvpBottom: una fila o columna de total sigue a los hijos, con la etiqueta *<miembro> Total*.
TsgcPivotField.Subtotals	False oculta los subtotales de los miembros de ese campo. Por defecto True.
TsgcPivotEngine.GrandTotalCaption	El título de los totales generales, Grand Total por defecto.
TsgcPivotEngine.BlankCaption	La etiqueta del miembro de los valores vacíos, (blank) por defecto.

- Los totales se calculan a partir de las filas de origen con combinaciones exactas, de modo que un promedio o un recuento de distintos de un total es correcto. Consulte Agregaciones.
- Después de un filtro de valor o de Top N, los totales incluyen solo los miembros que siguen visibles, como en Excel.
- Las celdas del total general siempre se calculan, también cuando la fila o la columna del total general está oculta, de modo que mostrar valores como sigue teniendo sus valores base.
- El renderizador y las cuadrículas dibujan los totales en negrita mientras TotalsBold es True, y las reglas de formato los seleccionan con los ámbitos pveTotals y pveGrandTotals.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure ConfigureTotals(aEngine: TsgcPivotEngine);
var
  oRegion: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    oRegion := aEngine.AddField('Region', pvaRow);
    aEngine.AddField('Product', pvaRow);
    aEngine.AddField('Channel', pvaColumn);
    aEngine.AddDataField('Sales', pvfSum);

    // subtotals under their members, a grand total row, no grand total column
    aEngine.SubtotalPosition := pvpBottom;
    aEngine.RowGrandTotals := True;
    aEngine.ColumnGrandTotals := False;
    aEngine.GrandTotalCaption := 'All regions';
    aEngine.BlankCaption := '(no value)';

    // the regions show no subtotals, only their products
    oRegion.Subtotals := False;
  finally
    aEngine.EndUpdate;
```

```
  end;
end;
```

Véase también

- Filtros
- TsgcPivotEngine

Filtros

Cuatro filtros reducen lo que muestra una tabla dinámica. Los tres primeros son propiedades de un campo, el cuarto es un evento del motor.

Filtro	Se aplica a	Cómo funciona
Filtro de miembros: FilterValues, FilterExclude	Campos de fila, columna y filtro	FilterValues enumera etiquetas de miembro, comparadas distinguiendo mayúsculas, tal como se muestran (después de DateGrouping y DisplayFormat). Las filas de origen deben tener una de ellas, o ninguna con FilterExclude. Vacío conserva todas las filas.
LabelFilter	Campos de fila, columna y filtro	Operator, Value y Value2 comparados con la etiqueta del miembro, sin distinguir mayúsculas. Cuando ambos lados son números se comparan como números.
ValueFilter	Campos de fila y de columna	Operator, Value y Value2 comparados con el total de cada miembro para el campo de datos cuyo AreaIndex es Measure. Se ejecuta nivel por nivel, dentro de cada elemento padre. Un total vacío solo pasa con pvcNone.
OnFilterRow	Filas de origen	Se genera para cada fila de origen mientras está asignado. Establezca aAccept a False para excluir la fila.

Los filtros de miembro y de etiqueta eliminan filas de origen, de modo que cambian todos los totales. El filtro de valor elimina miembros después de la agregación, y los totales se calculan de nuevo sobre los miembros que quedan.

Operadores

Operador	Coincide cuando
pvcNone	Siempre: sin filtro.
pvcEqual, pvcNotEqual	El valor es igual a Value, o no lo es.
pvcGreater, pvcGreaterEqual, pvcLess, pvcLessEqual	El valor se compara con Value.
pvcBetween, pvcNotBetween	El valor está entre Value y Value2, ambos incluidos, en cualquier orden, o no lo está.
pvcContains, pvcNotContains, pvcBeginsWith, pvcEndsWith	Pruebas de texto de la etiqueta (LabelFilter) o del texto invariante del valor.

Listas de miembros

TsgcPivotEngine.GetMemberLabels(aField, aList) rellena una lista con las etiquetas con las que se compara FilterValues, en orden de visualización. El menú emergente de miembros de las listas de campos lo usa: OK guarda la lista más corta, los miembros marcados, o los desmarcados con FilterExclude, y marcar todos los miembros borra el filtro.

Ejemplo

```
uses
  SysUtils, Variants,
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

type
  TOrderFilter = class
  private
    FQuantityCol: Integer;
  public
    procedure Install(aEngine: TsgcPivotEngine);
    procedure FilterRow(Sender: TObject; aRow: Integer; var aAccept: Boolean);
  end;

procedure TOrderFilter.Install(aEngine: TsgcPivotEngine);
var
  i: Integer;
begin
  FQuantityCol := -1;
  for i := 0 to aEngine.SourceColumnCount - 1 do
    if SameText(aEngine.SourceColumnName(i), 'Quantity') then
      FQuantityCol := i;
  aEngine.OnFilterRow := FilterRow;
  aEngine.Recalculate;
end;

procedure TOrderFilter.FilterRow(Sender: TObject; aRow: Integer;
  var aAccept: Boolean);
var
  vQuantity: Variant;
begin
  // a rule per source row: only the orders with a quantity above zero
  if FQuantityCol < 0 then
    Exit;
  vQuantity := TsgcPivotEngine(Sender).SourceValue(aRow, FQuantityCol);
  aAccept := not VarIsNull(vQuantity) and (vQuantity > 0);
end;

procedure ApplyFilters(aEngine: TsgcPivotEngine);
var
  oField: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    // a report filter: only two countries
    oField := aEngine.AddField('Country', pvaFilter);
    oField.FilterValues.Add('Spain');
    oField.FilterValues.Add('France');

    // every product except the samples
    oField := aEngine.AddField('Product', pvaRow);
    oField.FilterValues.Add('Sample');
    oField.FilterExclude := True;

    // a label filter: the customers whose name begins with A
    oField := aEngine.AddField('Customer', pvaRow);
    oField.LabelFilter.Operator := pvcBeginsWith;
    oField.LabelFilter.Value := 'A';

    // a value filter: the channels that sold more than 10,000
    oField := aEngine.AddField('Channel', pvaColumn);
    oField.ValueFilter.Operator := pvcGreater;
    oField.ValueFilter.Value := 10000;
    oField.ValueFilter.Measure := 0;

    aEngine.AddDataField('Sales', pvfSum);
```

```
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- Top N
- Totales
- TsgcPivotField

Top N

TopN conserva, dentro de cada padre, los N miembros de un campo de fila o de columna con el mayor total. Es una función avanzada, no incluida en la edición Community.

Propiedad	Efecto
TsgcPivotField.TopN	El número de miembros que se conservan. 0 (por defecto) conserva todos los miembros.
TsgcPivotField.SortMeasure	El campo de datos, por su AreaIndex, cuyos totales clasifican los miembros. También es la medida de una ordenación por valor.
TsgcPivotField.TopNOthers	True añade un miembro Others con el total de los miembros excluidos. Siempre va al final.
TsgcPivotEngine.OthersCaption	La etiqueta del miembro Others, Others por defecto.

- Top N decide qué miembros permanecen. SortBy y SortOrder deciden su orden, de modo que pvsValue con pvoDescending los enumera del mayor al menor.
- Los totales de los padres y los totales generales incluyen solo los miembros conservados, más Others cuando se muestra.
- Un drill-down sobre Others devuelve las filas de origen de todos los miembros que contiene.
- Un filtro de valor nunca crea un miembro Others.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure TopProducts(aEngine: TsgcPivotEngine);
var
  oProduct: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    oProduct := aEngine.AddField('Product', pvaRow);
    aEngine.AddField('Date', pvaColumn, pvdYear);
    aEngine.AddDataField('Revenue', pvfSum);

    // the 5 products with the largest revenue, the rest in one Others row
    oProduct.TopN := 5;
    oProduct.TopNOthers := True;
    oProduct.SortBy := pvsValue;
    oProduct.SortOrder := pvoDescending;
    oProduct.SortMeasure := 0;
    aEngine.OthersCaption := 'Other products';
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- Filtros
- Ediciones

Mostrar valores como

ShowAs cambia cómo se muestran los valores de un campo de datos, sin cambiar los valores en sí: RawValue, los totales, la ordenación, los filtros y el drill-down siguen usando los números reales. Es una función avanzada, no incluida en la edición Community.

ShowAs	Valor mostrado
pvwValue	El propio valor. El valor por defecto.
pvwPercentOfRow	El valor dividido por el total de su fila, como porcentaje.
pvwPercentOfColumn	El valor dividido por el total de su columna, como porcentaje.
pvwPercentOfGrandTotal	El valor dividido por el total general, como porcentaje.
pvwRunningTotal	A lo largo de ShowAsAxis, el valor más los valores de los miembros anteriores con el mismo padre, en orden de visualización.
pvwDifferenceFrom	El valor menos el valor del miembro base con el mismo padre: el miembro anterior cuando ShowAsBaseItem está vacío, si no el miembro con la etiqueta ShowAsBaseItem (sin distinguir mayúsculas).
pvwRank	La posición entre los miembros con el mismo padre, el mayor es 1. Los empates comparten posición y la siguiente se salta: 1, 2, 2, 4.

- Un modo de porcentaje muestra 25 para el 25 por ciento, con formato `0.00%`. Un valor vacío, una base ausente y una base de 0 dan una celda vacía.
- ShowAsAxis (pvxColumns por defecto) es el eje a lo largo del cual se toman el total acumulado, la diferencia y la posición.
- El primer miembro de un grupo no tiene nada con qué compararse en pvwDifferenceFrom, y el propio miembro base está vacío, como en Excel. El total general de ShowAsAxis no tiene hermanos, de modo que queda vacío en los tres últimos modos.
- La misma columna de origen puede añadirse dos veces, una como valor y otra con ShowAs, para mostrar ambos.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure CompareYears(aEngine: TsgcPivotEngine);
var
  oField: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    aEngine.AddField('Region', pvaRow);
    aEngine.AddField('Date', pvaColumn, pvdYear);
    aEngine.AddDataField('Sales', pvfSum);

    // the same measure as a share of the total of its year
    oField := aEngine.AddDataField('Sales', pvfSum);
    oField.Caption := '% of year';
    oField.ShowAs := pvwPercentOfColumn;

    // and as the change against the previous year
```

```
    oField := aEngine.AddDataField('Sales', pvfSum);
    oField.Caption := 'Change';
    oField.ShowAs := pvwDifferenceFrom;
    oField.ShowAsAxis := pvxColumns;
    oField.ShowAsBaseItem := '';
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- Agregaciones
- Ediciones

Campos calculados

Un campo calculado es una columna numérica calculada para cada fila de origen a partir de una expresión. Los campos lo usan mediante FieldName exactamente como una columna de origen. Es una función avanzada, no incluida en la edición Community.

Nivel de fila

La expresión se evalúa por fila de origen y después se agrega, de modo que `Profit = [Revenue] - [Cost]` sumado es la suma de los beneficios. No es la fórmula sobre los totales que calcula Excel: el promedio de un margen por fila no es el margen de los totales. Para una razón de totales, muestre ambas medidas y calcule la razón en OnGetCellText, o use mostrar valores como.

Sintaxis

Elemento	Sintaxis
Números	Decimales, con punto y un exponente opcional: 12, 0.5, 1E3.
Fields	[Field Name] o un nombre simple de letras, dígitos y guiones bajos, comparado con las columnas de origen y los demás campos calculados, sin distinguir mayúsculas.
Operadores	De menor a mayor precedencia: `OR` ; `AND` ; `NOT` ; `= <> < <= > >=` ; `+ -` ; `* /` ; `-` y `+` unarios; `^` (potencia, asociativa por la derecha). Los paréntesis agrupan.
Funciones	IF(c, a, b), ABS(x), ROUND(x, n), MIN(x, ...), MAX(x, ...), SQRT(x), ISNULL(x, v), YEAR(d), MONTH(d), sin distinguir mayúsculas.

- Un operando vacío da un resultado vacío, salvo en ISNULL, que devuelve v cuando x está vacío, y en IF, que solo mira la condición y la rama que toma.
- La división por cero, la raíz cuadrada de un número negativo y un desbordamiento dan un resultado vacío.
- Las comparaciones y los operadores lógicos devuelven 1 o 0. Cualquier valor distinto de 0 es verdadero.
- ROUND redondea la mitad alejándose de cero, n va de -15 a 15.

Errores

Un error de sintaxis o una referencia circular entre campos calculados genera EsgcPivotError en el siguiente recálculo, con el nombre del campo calculado en el mensaje. Un nombre de campo que no está cargado no es un error: el campo calculado permanece vacío hasta que esa columna llegue, de modo que el diseño puede definirse antes que los datos. Una Expression vacía también da valores vacíos.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure AddProfit(aEngine: TsgcPivotEngine);
var
  oCalc: TsgcPivotCalculatedField;
begin
  aEngine.BeginUpdate;
  try
    oCalc := aEngine.CalculatedFields.Add;
```

```
    oCalc.Name := 'Profit';
    oCalc.Expression := '[Revenue] - [Cost]';

    // a calculated field can use another one
    oCalc := aEngine.CalculatedFields.Add;
    oCalc.Name := 'Margin';
    oCalc.Expression := 'IF([Revenue] > 0, ROUND([Profit] / [Revenue] * 100, 1), 0)';

    aEngine.AddField('Product', pvaRow);
    aEngine.AddDataField('Profit', pvfSum);
    aEngine.AddDataField('Margin', pvfAverage).DisplayFormat := '0.0';
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- TsgcPivotCalculatedField
- Agregaciones

Persistencia del diseño

El diseño de una tabla dinámica, sus campos, filtros, totales y miembros expandidos, se guarda y se carga como JSON. Es una función avanzada, no incluida en la edición Community.

Método	Descripción
SaveLayoutToString: string	El diseño como texto JSON.
LoadLayoutFromString(aJSON)	Sustituye el diseño con un solo recálculo.
SaveLayoutToFile(aFileName)	Escribe el JSON como archivo UTF-8.
LoadLayoutFromFile(aFileName)	Lee un archivo UTF-8 y sustituye el diseño.

TsgcPivotEngine y ambas cuadrículas tienen estos métodos. Las versiones de las cuadrículas también guardan los anchos de columna que cambió el usuario.

Formato

El documento es un solo objeto: `v` (la versión del formato, 1), `rowGrandTotals`, `columnGrandTotals`, `subtotalPosition`, `calculatedFields` (nombre y expresión), `fields` (un objeto por campo, agrupados por área, con todas las propiedades de TsgcPivotField), `expanded` (el estado expandido de los niveles y de las rutas de miembro de las filas y de las columnas) y, desde una cuadrícula, `columnWidths`. Las enumeraciones se escriben como el nombre del miembro sin su prefijo y con la primera letra en minúscula: pvfCountDistinct es `countDistinct`.

Reglas de carga

- La carga es tolerante: los miembros desconocidos se ignoran, y un miembro ausente o un valor de tipo incorrecto conserva el valor por defecto.
- Una vez que el motor tiene datos, se omite un campo cuyo nombre no es ni una columna de origen ni un campo calculado.
- Un diseño cargado antes que los datos conserva su estado expandido hasta que los campos coinciden con los datos.
- Un JSON mal formado, o una raíz que no es un objeto, genera EsgcPivotError y deja el diseño sin cambios.

Ejemplo

```
uses
  SysUtils,
  sgcPivotGrid_Types, sgcPivotGrid_Engine, sgcPivotGrid_VCL;

procedure SaveUserLayout(aGrid: TsgcVCLPivotGrid; const aFileName: string);
begin
  // the fields, filters, totals, expanded members and column widths
  aGrid.SaveLayoutToFile(aFileName);
end;

function RestoreUserLayout(aGrid: TsgcVCLPivotGrid;
  const aFileName: string): Boolean;
begin
  Result := FileExists(aFileName);
  if Result then
  try
    aGrid.LoadLayoutFromFile(aFileName);
  except
```

```
    on EsgcPivotError do
      Result := False; // malformed file, the layout is unchanged
  end;
end;

procedure CopyLayout(aFrom, aTo: TsgcPivotEngine);
begin
  aTo.LoadLayoutFromString(aFrom.SaveLayoutToString);
end;
```

Véase también

- Áreas
- Ediciones

Exportación

El motor y ambas cuadrículas exportan la vista actual, con sus totales y su estado expandido. Cada método tiene una sobrecarga de archivo y otra de flujo, y el texto se escribe como UTF-8 sin BOM.

Método	Salida
SaveToCSV	CSV según RFC 4180 con registros CRLF. Una columna por nivel de fila, una fila de encabezado por nivel de columna más la fila de medidas, y una etiqueta de encabezado repetida en cada celda que cubre. Los valores se escriben como números invariantes, 1250.5, no con el formato de visualización, y una celda vacía es un campo vacío.
SaveToHTML	Un documento HTML completo con una tabla. Los encabezados combinados usan colspan y rowspan, y los valores son los textos formateados de las celdas, con el EmptyText del motor en una celda vacía.
SaveToJSON	La instantánea estructurada: los títulos de nivel, las medidas con sus formatos, y los elementos de columna y de fila con su tipo (member, total o grandTotal), nivel, etiqueta, ruta y estado expandido. Un elemento de fila también tiene sus valores después de mostrar valores como, null cuando está vacío.
SaveToPNG, SaveToSVG	Solo las cuadrículas y el renderizador: toda la cuadrícula como imagen. El ancho y el alto están en DIP, 0 es el tamaño del contenido, hasta 16384 píxeles.

Como tabla

TsgcPivotEngine y ambas cuadrículas implementan IsgcUITable mediante ResultTable, de modo que el resultado alimenta a cualquier consumidor de sgcUI, como un origen de datos de gráfico de sgcCharts u otra tabla dinámica. Las filas son las hojas de fila visibles, con los miembros contraídos incluidos y las filas de total excluidas. Las columnas son una columna de texto por nivel de fila y después una columna por hoja de columna visible y medida, con el nombre *<ruta de columna>* | *<título de la medida>*, o solo el título de la medida cuando no hay campos de columna. Una celda de valor vacía tiene el EmptyText del motor como TableText y Null como TableValue. TableVersion cambia con cada nueva instantánea. Consulte Interoperabilidad con gráficos.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPivotGrid_VCL;

procedure ExportView(aGrid: TsgcVCLPivotGrid; const aFolder: string);
var
  vPath: string;
  oStream: TMemoryStream;
begin
  vPath := IncludeTrailingPathDelimiter(aFolder);

  // the view with its totals: CSV and HTML tables, the JSON snapshot
  aGrid.SaveToCSV(vPath + 'pivot.csv');
  aGrid.SaveToHTML(vPath + 'pivot.html');
  aGrid.SaveToJSON(vPath + 'pivot.json');

  // the whole grid as an image, 0 is the content size
  aGrid.SaveToPNG(vPath + 'pivot.png');
  aGrid.SaveToSVG(vPath + 'pivot.svg', 1200, 800);
```

```
  // every export also writes to a stream
  oStream := TMemoryStream.Create;
  try
    aGrid.SaveToCSV(oStream);
  finally
    oStream.Free;
  end;
end;
```

Véase también

- Inicio rápido sin interfaz
- TsgcPivotGridRenderer

Expandir y contraer

Todo miembro que tiene hijos, en las filas o en las columnas, puede expandirse para mostrarlos o contraerse a una sola fila o columna. Un miembro contraído conserva su lugar y muestra los totales de todo lo que tiene debajo.

En las cuadrículas

- Los botones + y - de los encabezados de fila y de columna alternan un miembro. ShowExpandButtons los oculta.
- Un doble clic en un encabezado de fila alterna su miembro.
- Con el encabezado de fila enfocado, Derecha expande, Izquierda contrae y Enter alterna. Las teclas + y - expanden y contraen el miembro de la fila seleccionada desde cualquier columna. Consulte Teclado y accesibilidad.
- OnExpandChange se genera después de que el usuario expandiera o contrajera un miembro, con aAxis y aItem. No se genera para los métodos siguientes.

En código

Miembro	Efecto
TsgcPivotEngine.ExpandAll(aAxis), CollapseAll(aAxis)	Todos los miembros de pvxRows o pvxColumns.
TsgcPivotEngine.ExpandLevel(aAxis, aLevel), CollapseLevel(aAxis, aLevel)	Todos los miembros de un nivel, 0 es el nivel más externo.
TsgcPivotEngine.SetExpanded(aAxis, aItem, aValue), ToggleExpanded(aAxis, aItem)	Un elemento de la instantánea actual, por su índice en Result.Item. Un elemento sin hijos se deja como está.
TsgcPivotField.Expanded	El estado inicial de los miembros del nivel de ese campo. Por defecto True. Consulte Agrupación de fechas.
TsgcPivotEngine.GetExpandState(aAxis, aState), SetExpandState(aAxis, aState)	El estado expandido explícito de un eje como lista de cadenas, por ejemplo para copiarlo a otro motor. Edición Advanced.
TsgcPivotItem.Expandable, Expanded	Se leen de Result.Item: el elemento tiene hijos, y se muestran.

Cómo se conserva el estado

- El estado se guarda por ruta de miembro, como North / Spain, no por índice de elemento. Sobrevive a un recálculo, a una recarga de los datos, a un filtro y a un nuevo orden, y un miembro que desaparece y vuelve conserva su estado.
- Cuando cambian los campos de un eje (otro campo, otro orden u otra agrupación de fechas), el estado de ese eje empieza de nuevo desde la propiedad Expanded de sus campos.
- Expandir o contraer construye una nueva instantánea sobre los datos calculados sin volver a agregar, por lo que es rápido incluso con un resultado grande. Consulte la tabla de rendimiento en Arquitectura.
- Las exportaciones y ResultTable siguen el estado actual: un miembro contraído es una fila con sus totales.
- Persistencia del diseño guarda y carga el estado expandido junto con el diseño.

Ejemplo

```
uses
  Classes,
  sgcPivotGrid_Types, sgcPivotGrid_Engine;

procedure ShowOverview(aEngine: TsgcPivotEngine);
begin
  // rows Region > Country > City, columns Date by year > quarter
  aEngine.CollapseAll(pvxColumns); // only the years
  aEngine.CollapseAll(pvxRows);
  aEngine.ExpandLevel(pvxRows, 0); // the regions show their countries
end;

procedure ExpandRegion(aEngine: TsgcPivotEngine; const aRegion: string);
var
  i, vFound: Integer;
  vItem: TsgcPivotItem;
begin
  vFound := -1;
  aEngine.Lock;
  try
    if aEngine.Result <> nil then
      for i := 0 to aEngine.Result.ItemCount(pvxRows) - 1 do
      begin
        vItem := aEngine.Result.Item(pvxRows, i);
        if (vItem.Kind = pvkMember) and (vItem.Level = 0) and vItem.Expandable
          and (aEngine.Result.NodeLabel(pvxRows, vItem.Node) = aRegion) then
        begin
          vFound := i;
          Break;
        end;
      end;
  finally
    aEngine.Unlock;
  end;
  // after Unlock: the change publishes a new snapshot
  if vFound >= 0 then
    aEngine.SetExpanded(pvxRows, vFound, True);
end;

procedure CopyRowState(aFrom, aTo: TsgcPivotEngine);
var
  oState: TStringList;
begin
  // Advanced edition: the expanded state of the rows of another engine
  oState := TStringList.Create;
  try
    aFrom.GetExpandState(pvxRows, oState);
    aTo.SetExpandState(pvxRows, oState);
  finally
    oState.Free;
  end;
end;
```

SetExpanded se llama después de Unlock, porque el cambio publica una nueva instantánea.

Véase también

- Totales
- Persistencia del diseño
- TsgcPivotEngine

Ordenación

Cada campo de fila y de columna ordena sus miembros dentro de cada padre, de modo que los países de cada región se ordenan por separado. Tres propiedades de TsgcPivotField lo controlan.

Propiedad	Efecto
SortBy	pvsLabel (por defecto) ordena por la etiqueta del miembro. pvsValue ordena por el total de cada miembro.
SortOrder	pvoAscending (por defecto) o pvoDescending.
SortMeasure	El campo de datos, por su AreaIndex, cuyos totales compara una ordenación por valor. Por defecto 0. Top N también lo usa.

Reglas

- Una ordenación por etiqueta compara los miembros de texto sin distinguir mayúsculas, con las reglas de la configuración regional, y los miembros numéricos y de fecha por su valor, de modo que 9 va antes que 10 y 2024-10 después de 2024-09.
- Una ordenación por valor compara el total de cada miembro para la medida SortMeasure: el total sobre todas las columnas para un campo de fila, y sobre todas las filas para un campo de columna. Los miembros con total vacío van al final, y los miembros con el mismo total conservan el orden de su etiqueta.
- Cuando SortMeasure no es el AreaIndex de un campo de datos, los miembros se ordenan por etiqueta.
- El miembro en blanco y el miembro Others de Top N permanecen al final en ambos órdenes, Others después del miembro en blanco.
- Top N elige primero los miembros con los mayores totales de SortMeasure, y después SortBy y SortOrder los ordenan.
- Un cambio de SortBy, SortOrder o SortMeasure recalcula la tabla dinámica, una vez por bloque BeginUpdate y EndUpdate.
- Las cuadrículas no ordenan al hacer clic en un encabezado. Para que el usuario pueda ordenar, establezca las propiedades en código, por ejemplo en OnCellClick cuando la Part del impacto sea pvhColumnHeader.

Ejemplo

```
uses
  sgcPivotGrid_Types, sgcPivotGrid_Fields, sgcPivotGrid_Engine;

procedure SortBySales(aEngine: TsgcPivotEngine);
var
  oRegion, oProduct, oYear: TsgcPivotField;
begin
  aEngine.BeginUpdate;
  try
    aEngine.ClearLayout;
    oRegion := aEngine.AddField('Region', pvaRow);
    oProduct := aEngine.AddField('Product', pvaRow);
    oYear := aEngine.AddField('Date', pvaColumn, pvdYear);
    aEngine.AddDataField('Quantity', pvfSum); // measure 0
    aEngine.AddDataField('Sales', pvfSum); // measure 1
    // the regions by their sales total, the largest first
    oRegion.SortBy := pvsValue;
    oRegion.SortMeasure := 1;
    oRegion.SortOrder := pvoDescending;
    // the products of each region by quantity, the smallest first
    oProduct.SortBy := pvsValue;
    oProduct.SortMeasure := 0;
    // the newest year first: dates sort by date, not by label
```

```
    oYear.SortOrder := pvoDescending;
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- Top N
- Agrupación de fechas
- TsgcPivotField

Drill-down

Un drill-down lee las filas de origen que hay detrás de una celda: las filas que se agregaron en su valor. El motor las devuelve como números de fila o como tabla.

Método	Devuelve	Descripción
TsgcPivotEngine.GetSourceRows(aRowItem, aDataCol, var aRows)	Integer	Los números de las filas de origen, en el orden de las filas, y su cantidad. aRowItem es un elemento de fila de Result y aDataCol una columna de datos. aDataCol -1 toma todas las columnas del elemento de fila.
TsgcPivotEngine.DrillDown(aRowItem, aDataCol)	TsgcPivotSourceView	Las mismas filas como un IsgcUITable con todas las columnas del almacén, también las columnas que no están en el diseño. El llamador lo libera.
TsgcPivotEngine.SourceValue(aRow, aCol)	Variant	Un valor de una fila de origen, Null cuando está vacío. SourceColumnName da los nombres de columna.

Qué filas

- Una celda de miembro devuelve las filas de su miembro de fila y de su miembro de columna. Un subtotal o un total general devuelve todas las filas que tiene debajo, y el miembro Others de Top N devuelve las filas de todos los miembros que contiene.
- Las filas excluidas por un filtro de miembro, un filtro de etiqueta, un filtro de informe o OnFilterRow nunca se devuelven.
- Todas las medidas de un elemento de columna tienen las mismas filas, de modo que la medida de aDataCol no cambia el resultado.

En las cuadrículas

Enter sobre una celda de datos y un doble clic sobre una celda de datos generan OnDrillDown con aRowItem, aDataCol y var aHandled. La cuadrícula no tiene acción propia, de modo que la aplicación muestra las filas en un diálogo, una cuadrícula o un informe. La demo DataSetPivot abre los pedidos que hay detrás de una celda.

Ciclo de vida

- TsgcPivotSourceView lee el almacén del motor en vivo. Después de que cambien los datos de origen no tiene filas, y debe liberarse antes que el motor.
- Los índices de elemento pertenecen a la instantánea actual. Tómelos de los eventos de la cuadrícula, o lea Result entre Lock y Unlock.

Ejemplo

```
uses
  Classes, SysUtils, Variants,
  sgcPivotGrid_Types, sgcPivotGrid_Engine, sgcPivotGrid_VCL;

procedure ListOrders(aEngine: TsgcPivotEngine; aRowItem, aDataCol: Integer;
  aList: TStrings);
var
  vRows: TsgcPivotIntegers;
```

```
  i, vCount, vOrderCol, vSalesCol: Integer;
begin
  aList.Clear;
  vOrderCol := -1;
  vSalesCol := -1;
  for i := 0 to aEngine.SourceColumnCount - 1 do
    if SameText(aEngine.SourceColumnName(i), 'Order') then
      vOrderCol := i
    else if SameText(aEngine.SourceColumnName(i), 'Sales') then
      vSalesCol := i;
  if (vOrderCol < 0) or (vSalesCol < 0) then
    Exit;
  // the source rows behind the cell, aDataCol -1 takes the whole row
  vCount := aEngine.GetSourceRows(aRowItem, aDataCol, vRows);
  for i := 0 to vCount - 1 do
    aList.Add(VarToStr(aEngine.SourceValue(vRows[i], vOrderCol)) + ': ' +
      VarToStr(aEngine.SourceValue(vRows[i], vSalesCol)));
end;

type
  TDrillDownEvents = class
  public
    Lines: TStrings;
    procedure DrillDown(Sender: TObject; aRowItem, aDataCol: Integer;
      var aHandled: Boolean);
  end;

procedure TDrillDownEvents.DrillDown(Sender: TObject;
  aRowItem, aDataCol: Integer; var aHandled: Boolean);
var
  oView: TsgcPivotSourceView;
  r, c: Integer;
  vLine: string;
begin
  // every store column of the rows behind the cell, as an IsgcUITable
  oView := TsgcVCLPivotGrid(Sender).Pivot.DrillDown(aRowItem, aDataCol);
  try
    Lines.Clear;
    for r := 0 to oView.TableRowCount - 1 do
    begin
      vLine := '';
      for c := 0 to oView.TableColCount - 1 do
      begin
        if c > 0 then
          vLine := vLine + #9;
        vLine := vLine + oView.TableText(r, c);
      end;
      Lines.Add(vLine);
    end;
  finally
    oView.Free;
  end;
  aHandled := True;
end;
```

Véase también

- TsgcPivotEngine
- TsgcVCLPivotGrid
- Filtros

Formato condicional

La colección FormatRules de las cuadrículas y del renderizador colorea las celdas de datos según sus valores. Las mismas reglas se aplican en pantalla y en las imágenes PNG y SVG, porque las cuadrículas y las imágenes las dibuja el mismo renderizador. En tiempo de diseño las reglas se editan con el editor de colecciones del Object Inspector.

Kind	Qué hace
pvrValue	Una regla de valor. Cuando el valor cumple Operator, Value y Value2 (los operadores de los filtros), la celda recibe BackColor, ForeColor y Bold. pvcNone coincide con todo valor que no esté vacío.
pvrColorScale	Una escala de color. Las celdas reciben un color entre MinColor y MaxColor, según su lugar entre el valor más pequeño y el más grande de las celdas de miembro de la medida. Operator, Value y Value2 seleccionan los valores que colorea.

Qué celdas

- Measure selecciona el campo de datos por su AreaIndex, y -1 aplica la regla a todas las medidas.
- Scope selecciona los tipos de celda: pveCells (el valor por defecto, las celdas de miembro), pveTotals y pveGrandTotals.
- Una celda vacía no coincide con ninguna regla y nunca se colorea. Muestra el EmptyText del motor.

Orden de las reglas

- Las reglas se ejecutan en el orden de la colección. Cada regla activa que coincide aplica sus colores distintos de 0, de modo que una coincidencia posterior anula a una anterior, y Bold se acumula.
- StopOnMatch termina la evaluación para esa celda, de modo que gana la primera regla que coincide. El ejemplo lo usa para tres bandas.
- OnGetCellStyle se ejecuta después de las reglas con var aBack, aFore y aBold, y puede cambiar cualquiera de ellos. Dentro de él, Renderer.CurrentResult es la instantánea que se está dibujando.

Colores

Los colores son valores TsgcUIColor en formato $AARRGGBB, por lo que $FF2E7D32 es un verde opaco, y 0 conserva el color del tema. Con el tema oscuro, elija colores que mantengan legible el texto, y combine el color con Bold para que el significado no dependa solo del color.

Ejemplo

```
uses
  sgcUICore_Canvas, sgcPivotGrid_Types, sgcPivotGrid_Result,
  sgcPivotGrid_Format, sgcPivotGrid_VCL;

procedure AddTrafficLights(aGrid: TsgcVCLPivotGrid);
var
  oRule: TsgcPivotFormatRule;
begin
  aGrid.FormatRules.Clear;
  // green from 10,000: StopOnMatch makes the first match win
  oRule := aGrid.FormatRules.Add;
  oRule.Measure := 0;
  oRule.Operator := pvcGreaterEqual;
  oRule.Value := 10000;
  oRule.BackColor := $FFC8E6C9;
```

```
  oRule.StopOnMatch := True;
  // amber from 5,000
  oRule := aGrid.FormatRules.Add;
  oRule.Measure := 0;
  oRule.Operator := pvcGreaterEqual;
  oRule.Value := 5000;
  oRule.BackColor := $FFFFECB3;
  oRule.StopOnMatch := True;
  // red for every other value, in bold so that it does not rely on colour
  oRule := aGrid.FormatRules.Add;
  oRule.Measure := 0;
  oRule.Operator := pvcNone;
  oRule.BackColor := $FFFFCDD2;
  oRule.Bold := True;
end;

type
  TStyleEvents = class
  public
    procedure GetCellStyle(Sender: TObject; aRowItem, aDataCol: Integer;
      var aBack, aFore: TsgcUIColor; var aBold: Boolean);
  end;

procedure TStyleEvents.GetCellStyle(Sender: TObject;
  aRowItem, aDataCol: Integer; var aBack, aFore: TsgcUIColor;
  var aBold: Boolean);
var
  oResult: TsgcPivotResult;
  vItem: TsgcPivotItem;
begin
  // the snapshot being drawn, valid while the event runs
  oResult := TsgcVCLPivotGrid(Sender).Renderer.CurrentResult;
  if (oResult = nil) or (oResult.MeasureCount = 0) then
    Exit;
  // the grand total column in blue text
  vItem := oResult.Item(pvxColumns, aDataCol div oResult.MeasureCount);
  if vItem.Kind = pvkGrandTotal then
    aFore := $FF1565C0;
end;
```

Véase también

- TsgcPivotFormatRule
- TsgcPivotGridRenderer
- Teclado y accesibilidad

Interoperabilidad con gráficos

sgcPivotGrid intercambia datos con los demás productos sgcUI mediante IsgcUITable, la interfaz de tabla de sgcUICore en la unidad sgcUICore_Table. Los paquetes de sgcPivotGrid no dependen de sgcCharts ni de sgcGrid, los conecta la aplicación.

De la tabla dinámica a sgcCharts

TsgcPivotEngine y ambas cuadrículas implementan IsgcUITable mediante ResultTable. Asigne la tabla dinámica a la propiedad Table de un TsgcChartDataSource_Table de sgcCharts, o llame a su SetTable con un motor que no tenga cuadrícula.

Parte de la tabla	Contenido
Filas	Las hojas de fila visibles: incluidos los miembros contraídos, excluidas las filas de subtotal y de total general.
Columnas de texto	Una por nivel de fila, con el nombre del título del nivel, con la etiqueta del miembro en ese nivel.
Columnas de valor	Una por hoja de columna visible y medida, con el nombre *<ruta de columna> | <título de la medida>*, por ejemplo `Online | Sales` , o solo el título de la medida cuando no hay campos de columna.
TableValue	La etiqueta de una columna de texto, y el valor mostrado de una columna de valor como Double (después de mostrar valores como), o Null cuando la celda está vacía.
TableText	La etiqueta, o el texto formateado de la celda. Una celda vacía da el EmptyText del motor.

- Asigne las columnas en el origen de datos: YColumn nombra la columna de valor, LabelColumn el nivel de fila de un gráfico circular, de embudo o treemap (varios niveles separados por ;), y XColumn los valores X de una serie XY.
- Dé al campo de datos un Caption, como Sales, para que los nombres de columna no cambien con la agregación.
- Cada nueva instantánea cambia TableVersion y notifica a los oyentes, de modo que el gráfico se recarga tras un recálculo, una expansión o una contracción.
- La demo GridToPivotToChart dibuja barras agrupadas con un origen de datos por canal, y rellena las categorías del eje X con las etiquetas de fila en OnCalculated.

De sgcGrid a la tabla dinámica

TsgcPivotSource_Table carga cualquier IsgcUITable en el motor: un TsgcVCLGrid, un TsgcFMXGrid o un TsgcGridData de sgcGrid. Se recarga cuando TableVersion cambia, de modo que una edición en la cuadrícula actualiza la tabla dinámica y después el gráfico. Agrupe muchos cambios de la cuadrícula en BeginUpdate y EndUpdate de los datos de la cuadrícula, lo que da una sola recarga.

De una tabla dinámica a otra

El resultado de una tabla dinámica también es una tabla, por lo que TsgcPivotSource_Table puede cargarlo en una segunda tabla dinámica.

Lectura de la tabla en código

Lea la tabla entre TableLock y TableUnlock. El recuento de referencias está desactivado, el motor posee la tabla, así que suelte toda referencia de interfaz antes de liberar el motor.

Ejemplo

```
uses
  Classes,
  sgcGrid_VCL, sgcPivotGrid_Types, sgcPivotGrid_Fields,
  sgcPivotGrid_Source_Table, sgcPivotGrid_VCL, sgcCharts_Chart,
  sgcCharts_Series_Pie, sgcCharts_DataSource_Table, sgcCharts_VCL;

procedure LinkGridPivotChart(aOwner: TComponent; aGrid: TsgcVCLGrid;
  aPivot: TsgcVCLPivotGrid; aView: TsgcVCLChart);
var
  oField: TsgcPivotField;
  oSource: TsgcPivotSource_Table;
  oChart: TsgcChart;
  oPie: TsgcChartSeries_Pie;
  oData: TsgcChartDataSource_Table;
begin
  // the layout: one row per region with the sum of the sales
  aPivot.Pivot.BeginUpdate;
  try
    aPivot.Pivot.ClearLayout;
    aPivot.Pivot.AddField('Region', pvaRow);
    oField := aPivot.Pivot.AddDataField('Sales', pvfSum);
    oField.Caption := 'Sales';
  finally
    aPivot.Pivot.EndUpdate;
  end;

  // sgcGrid to pivot: the grid implements IsgcUITable
  oSource := TsgcPivotSource_Table.Create(aOwner);
  oSource.Table := aGrid;
  oSource.PivotGrid := aPivot;
  oSource.Active := True;

  // pivot to sgcCharts: one pie slice per region
  oChart := TsgcChart.Create(aOwner);
  oPie := TsgcChartSeries_Pie.Create(aOwner);
  oPie.Chart := oChart;
  oData := TsgcChartDataSource_Table.Create(aOwner);
  oData.Series := oPie;
  oData.LabelColumn := 'Region';
  oData.YColumn := 'Sales';
  oData.Table := aPivot;
  aView.Chart := oChart;
end;
```

Véase también

- Exportación
- TsgcPivotSource_Table
- Demos

Teclado y accesibilidad

Ambas cuadrículas pueden usarse solo con el teclado. Las mismas teclas funcionan en TsgcVCLPivotGrid y en TsgcFMXPivotGrid, donde Cmd en macOS funciona como Ctrl.

Tecla	Acción
Arriba, Abajo	La fila anterior o la siguiente.
Re Pág, Av Pág	Una página hacia arriba o hacia abajo.
Inicio, Fin	La primera o la última columna de datos de la fila. Con Ctrl, la primera o la última celda de la cuadrícula.
Izquierda	La columna de datos anterior. Desde la primera columna de datos pasa al encabezado de fila, y en el encabezado de fila contrae el miembro.
Derecha	En el encabezado de fila expande un miembro contraído, y si no pasa a la primera columna de datos. En los datos pasa a la columna siguiente.
+, -	En el teclado numérico o en el principal: expande o contrae el miembro de la fila seleccionada, desde cualquier columna.
Intro	En una celda de datos genera OnDrillDown. En el encabezado de fila alterna el miembro.

- Cuando no hay nada seleccionado, la primera tecla de navegación selecciona la primera celda.
- TsgcVCLPivotGrid tiene TabStop True por defecto y TsgcFMXPivotGrid tiene CanFocus True. Un clic da el foco a la cuadrícula.
- En código, SelectedRow, SelectedCol y ScrollToCell mueven la selección, y OnSelectionChange informa de cada cambio.
- Las teclas expanden y contraen los miembros de fila. Los miembros de columna se expanden con los botones de los encabezados de columna o en código, consulte Expandir y contraer.
- Las listas de campos se usan con el ratón o con el tacto. Escape cancela un arrastre, y TsgcFMXPivotFieldList también desplaza su lista con Arriba, Abajo, Re Pág, Av Pág, Inicio y Fin. Para los usuarios de teclado, ofrezca los diseños en código con AddField y MoveField, o como diseños guardados.

Accesibilidad

- Las cuadrículas y las listas de campos las dibuja sgcUICore y no publican su contenido a los lectores de pantalla mediante UI Automation ni MSAA. Para los usuarios de lectores de pantalla, ofrezca los datos como HTML: SaveToHTML escribe una tabla real con celdas de encabezado, colspan y rowspan. ResultTable también puede rellenar un control estándar.
- Los tamaños están en DIP y siguen el DPI del monitor. FontSize, RowHeight y HeaderHeight agrandan el texto y las filas.
- Theme selecciona el tema claro o el oscuro, y AccentColor sustituye el color de acento del tema.
- El formato condicional puede añadir Bold a un color, para que un valor no dependa solo del color.

Ejemplo

```
uses
  Classes, Controls,
  sgcPivotGrid_Types, sgcPivotGrid_VCL;

procedure FocusFirstCell(aGrid: TsgcVCLPivotGrid);
begin
  // the first data cell gets the selection and the grid the keyboard
```

```
  aGrid.SelectedRow := 0;
  aGrid.SelectedCol := 0;
  aGrid.ScrollToCell(0, 0);
  if aGrid.CanFocus then
    aGrid.SetFocus;
end;

// from the OnKeyDown of a form with KeyPreview: Ctrl+E expands every row,
// Ctrl+Shift+E collapses them
procedure HandlePivotShortcut(aGrid: TsgcVCLPivotGrid; var Key: Word;
  Shift: TShiftState);
begin
  if not aGrid.Focused or (Key <> Ord('E')) or not(ssCtrl in Shift) then
    Exit;
  if ssShift in Shift then
    aGrid.Pivot.CollapseAll(pvxRows)
  else
    aGrid.Pivot.ExpandAll(pvxRows);
  Key := 0;
end;
```

Véase también

- TsgcVCLPivotGrid
- Drill-down

Rendimiento

El motor está pensado para datos grandes: un almacén de columnas tipado, una pasada sobre las filas de origen por recálculo y dibujo virtual. El grupo de rendimiento de QA mide los escenarios principales frente a los presupuestos indicados en Arquitectura. Estas pautas mantienen una aplicación dentro de ellos.

Carga

- Cargue dentro de BeginLoad y EndLoad. El motor recalcula una sola vez, en el EndLoad más externo. Fuera de un bloque, cada AppendRow y DefineColumn recalcula de inmediato.
- Defina las columnas con DefineColumn(aName, aType). Sin tipo, lo decide el primer valor que no esté vacío, y una columna de enteros se convierte en columna de coma flotante cuando llega un valor fraccionario.
- TsgcPivotSource_DataSet recarga todo el conjunto de datos después de cada post mientras LiveUpdates es True. Para un lote de ediciones establezca LiveUpdates o Active a False y llame a Load al final.
- TsgcPivotSource_Table recarga solo cuando TableVersion cambia. Agrupe los cambios de una cuadrícula en su BeginUpdate y EndUpdate.

Diseño y vista

- Cambie el diseño dentro de BeginUpdate y EndUpdate para un solo recálculo. Para lotes más largos establezca AutoRecalculate a False y llame a Recalculate al final.
- DeferUpdate de las listas de campos mantiene pendientes los cambios del usuario hasta ApplyChanges.
- Expand, collapse, RowGrandTotals, ColumnGrandTotals, SubtotalPosition, RowLayout y EmptyText solo cambian la vista. El motor construye una nueva instantánea sobre los datos calculados sin volver a agregar.

Tamaño del resultado

- El número de celdas crece con los miembros de las filas multiplicados por los miembros de las columnas. Un campo con muchos miembros, como un número de pedido o una fecha por día, en ambos ejes los multiplica. Un diseño que necesite más de MaxCells celdas (5.000.000 por defecto) genera EsgcPivotError. Agrupe las fechas, mueva el campo a los filtros o use Top N.
- Cada campo de datos añade un valor por celda. CountDistinct guarda los valores distintos de cada celda, y StdDev y Var guardan más estado que Sum, de modo que cuestan más memoria y tiempo en diseños grandes.

Eventos

- OnFilterRow se ejecuta para cada fila de origen en cada recálculo. Busque los índices de columna una sola vez, fuera del evento, como hace el ejemplo de Filtros. Los filtros de miembro y de etiqueta se deciden una vez por miembro, por lo que cuestan menos.
- OnGetCellText y OnGetCellStyle se ejecutan para cada celda visible en cada fotograma. Manténgalos breves.
- Mantenga el bloqueo de Result solo mientras lo lee.

Ejemplo

```
uses
  SysUtils,
  sgcPivotGrid_Types, sgcPivotGrid_Engine;

const
  CS_REGIONS: array [0 .. 3] of string = ('North', 'South', 'East', 'West');

procedure LoadSales(aEngine: TsgcPivotEngine; aCount: Integer);
var
```

```
  i: Integer;
  vRegion: string;
  vDate: TDateTime;
  vSales: Double;
begin
  // one load bracket: one recalculation at EndLoad, not one per row
  aEngine.BeginLoad(True);
  try
    // fixed types: no inference from the first value
    aEngine.DefineColumn('Region', pvtText);
    aEngine.DefineColumn('Date', pvtDateTime);
    aEngine.DefineColumn('Sales', pvtFloat);
    for i := 0 to aCount - 1 do
    begin
      vRegion := CS_REGIONS[i mod 4];
      vDate := EncodeDate(2024, 1, 1) + (i mod 366);
      vSales := (i mod 100) * 1.5;
      aEngine.AppendRow([vRegion, vDate, vSales]);
    end;
  finally
    aEngine.EndLoad;
  end;
end;

procedure SetLayout(aEngine: TsgcPivotEngine);
begin
  // one recalculation for the whole layout, at EndUpdate
  aEngine.BeginUpdate;
  try
    aEngine.ClearLayout;
    aEngine.AddField('Region', pvaRow);
    aEngine.AddField('Date', pvaColumn, pvdQuarter);
    aEngine.AddDataField('Sales', pvfSum);
  finally
    aEngine.EndUpdate;
  end;
end;
```

Véase también

- Arquitectura
- TsgcPivotEngine
- Solución de problemas

TsgcPivotGridAIQuery

Unit: sgcPivotGridWS_AI
Package: sgcPivotGridWS (Delphi) y sgcPivotGridWSC (C++Builder), el puente con sgcWebSockets de sgcPivotGrid. Edición Registered, solo Windows (Win32 y Win64).
TsgcPivotGridAIQuery es un componente no visual que convierte una pregunta en lenguaje natural, como *top 3 countries by sales*, en el diseño de un pivote. Envía la pregunta y un catálogo de los campos de origen a un proveedor de IA mediante un **TsgcAIChat** de sgcWebSockets, valida la respuesta JSON contra los campos del motor y la aplica en un único recálculo. Las filas de origen nunca se envían. Consulte Pregunte a su tabla dinámica para ver qué se envía, cómo se valida la respuesta y el modo asíncrono.
Suelte un TsgcAIChat y un TsgcPivotGridAIQuery desde la página **SGC PivotGrid** de la paleta, asigne Chat y PivotGrid, y llame a Ask. El proveedor, el modelo y la clave de API son propiedades del chat: Provider, ChatOptions.Model y ChatOptions.ApiKey.

Propiedades

Propiedad	Tipo	Descripción
Active	Boolean	Con False, Ask y AskAsync devuelven False y no hacen nada. True por defecto.
Chat	TsgcAI_Chat	El chat que habla con el proveedor: un TsgcAIChat o cualquier descendiente de TsgcAI_Chat. Sin chat, Ask dispara OnError.
PivotGrid	TComponent	La tabla dinámica cuyo diseño se cambia: un TsgcVCLPivotGrid, un TsgcFMXPivotGrid o cualquier componente que implemente IsgcPivotEngineHost.
Engine	TsgcPivotEngine	Public. Un motor para usar sin tabla, por ejemplo en un servidor. Tiene prioridad sobre PivotGrid.
SendSampleValues	Boolean	Envía con el catálogo unos pocos valores de cada campo de texto, entero, fecha y booleano, para que el proveedor pueda reconocer los miembros de una pregunta como *online sales*. True por defecto.
MaxSampleValues	Integer	El número de valores de muestra por campo, limitado a 0..5. 3 por defecto.
Language	string	El idioma de la explicación, por ejemplo `Spanish` . Vacío, el valor por defecto, la escribe en el idioma de la pregunta.
ReplaceLayout	Boolean	True, el valor por defecto, pide un diseño nuevo. False envía además el diseño actual, de modo que una pregunta como *add the channel to the columns* lo modifica. La respuesta es el diseño nuevo completo en ambos casos.
Busy	Boolean	Public, solo lectura. True mientras se ejecuta una petición de AskAsync.
LastLayout	TsgcPivotAILayout	Public. El diseño aplicado por la última respuesta. Está vacío tras Undo.
LastExplanation	string	Public. La explicación de la última respuesta aplicada, una frase corta para una barra de estado.

Eventos

Evento	Tipo	Descripción
OnBeforeAsk	TsgcPivotAIBeforeAskEvent	`(Sender; var aPrompt: string; var aAllow: Boolean)` . Se dispara en el hilo principal antes de la petición, con el mensaje de usuario. Cambie aPrompt para añadir sus propias instrucciones, o ponga aAllow a False para cancelar la pregunta en silencio.
OnLayout	TsgcPivotAILayoutEvent	`(Sender; aLayout: TsgcPivotAILayout; var aAccept: Boolean)` . Se dispara con el diseño validado antes de aplicarlo. Inspeccione o edite aLayout, o ponga aAccept a False para conservar el diseño actual.
OnApplied	TNotifyEvent	Se dispara después de aplicar el diseño. LastLayout y LastExplanation contienen la respuesta.
OnError	TsgcPivotAIErrorEvent	`(Sender; const aError: string)` . Se dispara cuando no se aplicó nada: sin chat, sin motor, una pregunta vacía, una petición que sigue en curso, un error del proveedor, una respuesta sin objeto JSON, o una respuesta sin ningún campo de fila, columna o datos utilizable. El diseño queda como estaba.

Métodos

Método	Devuelve	Descripción
Ask(aQuestion)	Boolean	Consulta al proveedor, valida la respuesta y la aplica. Síncrono, solo en el hilo principal. True cuando se aplicó un diseño.
AskAsync(aQuestion)	Boolean	Lo mismo, con la petición en un hilo de trabajo. La respuesta se valida y se aplica en el hilo principal, y después se dispara OnApplied u OnError. False cuando Busy es True o la petición no arrancó.
Undo		Restaura el diseño anterior a la última respuesta aplicada. Solo se guarda un nivel.
BuildPrompt(aQuestion)	string	El mensaje de usuario que Ask envía para aQuestion: la pregunta, la línea del idioma y el catálogo de campos. Vacío cuando no hay motor. Úselo para mostrar al usuario lo que sale de la aplicación.
ValidateLayoutJSON(aJSON, aLayout)	Boolean	Extrae, analiza y valida una respuesta o un diseño guardado contra los campos del motor y lo deja en aLayout. False, con aLayout vacío, cuando no se encuentra ningún objeto JSON o no queda ningún campo de fila, columna o datos.
ResolveEngine	TsgcPivotEngine	Engine si está asignado, si no el motor de PivotGrid, si no nil.

Los descendientes pueden sobrescribir tres métodos virtuales protegidos: **DoAskProvider**, el único método que habla con el proveedor (una prueba puede devolver desde él una respuesta predefinida, como hace la demo AskYourPivot en su autotest), **DoBuildFields**, que construye el catálogo de campos (un campo omitido no puede enviarse ni aceptarse), y **DoApply**, que sustituye el diseño del motor.

TsgcPivotAILayout

El diseño de una respuesta, tras la validación. OnLayout lo recibe y LastLayout conserva el aplicado.

Miembro	Descripción
Rows, Columns	Los campos de eje en orden: FieldName y Grouping (TsgcPivotDateGrouping).
Data	Los campos de datos en orden: FieldName y Aggregation (TsgcPivotAggregation). El índice de una entrada es el índice de medida al que se refieren los filtros, las ordenaciones y Top N.
Filters	Tipo `pafMembers` (Members, Exclude), `pafLabel` (FilterOperator, Value, Value2) o `pafValue` (FilterOperator, NumValue, NumValue2, Measure).
Sorts	FieldName, SortBy, SortOrder y Measure.
TopN	FieldName, Count, Measure y Others.
Explanation	La explicación del proveedor, depurada.
Clear, IsEmpty, Assign	Vacía el diseño, comprueba si no hay entradas de fila, columna ni datos, copia otro diseño.
ToJSON	El diseño como un objeto JSON compacto con la forma del esquema, de modo que un diseño guardado vuelve a validarse.
ToText	Un resumen de una línea del diseño, para una barra de estado.

Ejemplo

Un asistente que posee el chat y la consulta, pregunta de forma síncrona y rechaza un diseño con más de dos campos de fila:

```
uses
  SysUtils, Classes,
  sgcAI, sgcAI_Chat, sgcPivotGrid_VCL, sgcPivotGridWS_AI;

type
  TPivotAssistant = class(TComponent)
  private
    FChat: TsgcAIChat;
    FQuery: TsgcPivotGridAIQuery;
    FLog: TStrings;
    procedure QueryLayout(Sender: TObject; aLayout: TsgcPivotAILayout;
      var aAccept: Boolean);
    procedure QueryApplied(Sender: TObject);
    procedure QueryError(Sender: TObject; const aError: string);
  public
    procedure Setup(aGrid: TsgcVCLPivotGrid; const aApiKey: string;
      aLog: TStrings);
    function Ask(const aQuestion: string): Boolean;
  end;

procedure TPivotAssistant.Setup(aGrid: TsgcVCLPivotGrid;
  const aApiKey: string; aLog: TStrings);
begin
  FLog := aLog;
  // the provider, the model and the key live on the chat
  FChat := TsgcAIChat.Create(Self);
  FChat.Provider := aicpOpenAI;
  FChat.ChatOptions.Model := 'gpt-4o-mini';
```

```
  FChat.ChatOptions.ApiKey := aApiKey;

  FQuery := TsgcPivotGridAIQuery.Create(Self);
  FQuery.Chat := FChat;
  FQuery.PivotGrid := aGrid;
  FQuery.MaxSampleValues := 3;
  FQuery.OnLayout := QueryLayout;
  FQuery.OnApplied := QueryApplied;
  FQuery.OnError := QueryError;
end;

function TPivotAssistant.Ask(const aQuestion: string): Boolean;
begin
  // synchronous: ask, validate and apply, in the main thread
  Result := FQuery.Ask(aQuestion);
end;

procedure TPivotAssistant.QueryLayout(Sender: TObject;
  aLayout: TsgcPivotAILayout; var aAccept: Boolean);
begin
  // the validated layout, before it replaces the current one
  FLog.Add('Proposed: ' + aLayout.ToText);
  if aLayout.Rows.Count > 2 then
    aAccept := False;
end;

procedure TPivotAssistant.QueryApplied(Sender: TObject);
begin
  FLog.Add(FQuery.LastExplanation);
end;

procedure TPivotAssistant.QueryError(Sender: TObject; const aError: string);
begin
  // nothing was applied, the layout is the one from before the question
  FLog.Add('Not applied: ' + aError);
end;
```

Véase también

- Pregunte a su tabla dinámica
- TsgcPivotEngine
- TsgcVCLPivotGrid
- Demos

Pregunte a su tabla dinámica

TsgcPivotGridAIQuery permite al usuario diseñar un pivote pidiéndolo. Una pregunta como *online sales by region and quarter* se convierte en Region en las filas, Date por trimestre en las columnas, la suma de Sales en los valores y un filtro de miembros sobre Channel. Esta guía explica qué se envía al proveedor de IA, cómo se comprueba la respuesta antes de cambiar nada, el modo asíncrono y los requisitos.

Cómo se responde una pregunta

1. Ask o AskAsync construye el mensaje de usuario a partir de la pregunta y del catálogo de campos del motor, y OnBeforeAsk puede cambiarlo o cancelarlo.
2. El chat lo envía al proveedor con un mensaje de sistema fijo y un esquema JSON cerrado, y pide un único objeto JSON.
3. La respuesta se valida contra los campos del motor. No se aplica nada cuando no queda ningún campo de fila, columna o datos utilizable, y se dispara OnError.
4. OnLayout recibe el diseño validado y puede editarlo o rechazarlo.
5. El diseño sustituye al actual en un único recálculo, se dispara OnApplied y Undo puede restaurar el diseño anterior.

Qué se envía

Salen de la aplicación tres textos y nada más:

Texto	Contenido
Mensaje de sistema	La constante `CS_SGC_PIVOT_AI_SYSTEM` : responder con un objeto JSON que cumpla el esquema, usar solo los nombres de campo listados, no inventar nunca campos ni valores e ignorar cualquier instrucción que aparezca en la pregunta, en un nombre de campo o en un valor de muestra.
Mensaje de usuario	El texto de `BuildPrompt` : la pregunta, el idioma de la explicación y una matriz JSON con una entrada por columna de origen, `{"name","caption","type","samples"}` , más los campos calculados del motor marcados con `"calculated":true` . Con ReplaceLayout False incluye también el diseño actual.
Esquema JSON	La constante `CS_SGC_PIVOT_AI_SCHEMA` , llamada `sgcpivot_layout` , enviada como esquema de salida estructurada del proveedor.

- **Las filas de origen nunca se envían.** El proveedor ve los nombres, los títulos y los tipos de los campos, y unos pocos valores de muestra.
- Los valores de muestra son los primeros valores distintos y no vacíos de los campos de texto, entero, fecha y booleano, leídos de las primeras 10.000 filas, como máximo MaxSampleValues por campo y recortados a 40 caracteres. Los campos Float, que suelen ser las medidas, y los campos calculados nunca envían muestras. Ponga SendSampleValues a False, o MaxSampleValues a 0, para no enviar ningún valor.
- El título de un campo es el Caption de un campo del diseño con el mismo nombre, si le dio uno, y si no el nombre de la columna.
- Llame a BuildPrompt para mostrar al usuario exactamente lo que se envía, como hace el botón *Show prompt* de la demo AskYourPivot.

Un motor sin interfaz con tres columnas, el mensaje de usuario que produce y un diseño guardado comprobado por el mismo validador:

```
uses
  SysUtils, Classes,
  sgcPivotGrid_Types, sgcPivotGrid_Engine, sgcPivotGridWS_AI;

procedure InspectPrompt(aLog: TStrings);
```

```
var
  oEngine: TsgcPivotEngine;
  oQuery: TsgcPivotGridAIQuery;
  oLayout: TsgcPivotAILayout;
begin
  oEngine := TsgcPivotEngine.Create;
  oQuery := TsgcPivotGridAIQuery.Create(nil);
  oLayout := TsgcPivotAILayout.Create;
  try
    oEngine.BeginLoad;
    try
      oEngine.DefineColumn('Region', pvtText);
      oEngine.DefineColumn('Date', pvtDateTime);
      oEngine.DefineColumn('Sales', pvtFloat);
      oEngine.AppendRow(['North', EncodeDate(2025, 1, 15), 1200.0]);
      oEngine.AppendRow(['South', EncodeDate(2025, 4, 2), 450.0]);
    finally
      oEngine.EndLoad;
    end;
    // headless: an engine instead of a pivot grid, no chat needed here
    oQuery.Engine := oEngine;
    oQuery.MaxSampleValues := 2;

    // the user message that Ask sends: the question, the language line
    // and the field catalog with sample values, never the source rows
    aLog.Add(oQuery.BuildPrompt('sales by region and quarter'));

    // a stored layout goes through the same validator as an answer
    if oQuery.ValidateLayoutJSON('{"rows":[{"field":"region","group":"none"}],'
      + '"columns":[{"field":"Date","group":"quarter"}],'
      + '"data":[{"field":"Sales","aggregation":"sum"}],"filters":[],'
      + '"sort":[],"top":[],"explanation":"Sales by region and quarter."}',
      oLayout) then
    begin
      aLog.Add(oLayout.ToText);
      aLog.Add(oLayout.ToJSON);
    end
    else
      aLog.Add('Rejected: no usable row, column or data field');
  finally
    oLayout.Free;
    oQuery.Free;
    oEngine.Free;
  end;
end;
```

La respuesta

El esquema es cerrado: cada objeto lista todas sus claves y no admite ninguna otra. Una respuesta tiene las claves `rows` y `columns` (field y group: none, year, quarter, month, week o day), `data` (field y aggregation: sum, count, average, min, max, countDistinct, stdDev o var), `filters` (un filtro de miembros, de etiqueta o de valor), `sort` (por etiqueta o por valor, ascendente o descendente), `top` (Top N con un miembro Others opcional) y `explanation`. Una medida es el índice de una entrada de `data`, empezando en 0.
Algunos proveedores rechazan el modo de salida estructurada. Cuando el error del proveedor menciona el esquema o el formato de respuesta, la consulta pregunta una vez más en modo JSON simple, con el esquema pegado en el mensaje de usuario. Los cuatro ajustes del chat que cambia (ResponseFormat, JSONSchema, JSONSchemaName y SystemMessage) se restauran y el historial del chat se borra después de cada pregunta, de modo que el chat puede compartirse con otro código.

Validación

La respuesta es una entrada no fiable. ValidateLayoutJSON es la frontera entre el proveedor y el motor, y toda respuesta pasa por ella antes de OnLayout:

- La respuesta se corta a 64 KB. Se busca primero el primer bloque de código delimitado y después el primer objeto JSON equilibrado, saltando las llaves dentro de cadenas.
- El nombre de un campo debe coincidir exactamente con un campo del motor, sin distinguir mayúsculas y sin los espacios exteriores. Los campos calculados se aceptan solo en los valores. Los campos, claves y valores de enumeración desconocidos se descartan en silencio.
- Una agregación numérica de un campo de texto o booleano pasa a ser Count, y la explicación lo indica. Una agrupación de fechas solo se conserva en campos de fecha.
- Como máximo 4 entradas de fila, 4 de columna, 6 de datos, 8 de filtro, 4 de ordenación y 2 de Top N, 50 miembros por filtro, y N entre 1 y 100.
- Se eliminan los caracteres de control, los textos se cortan a 200 caracteres y la explicación a 300, sin corchetes angulares.
- Un campo en las filas o en las columnas no se pone además en el área de filtros. Un filtro de miembros de un campo que no está en ningún eje va al área de filtros.
- Una respuesta con campos de eje pero sin datos recibe el Count del primer campo de fila. Una respuesta sin ningún campo de fila, columna o datos se rechaza entera: no cambia nada y se dispara OnError.

Como el mismo método valida un diseño guardado, una aplicación puede guardar LastLayout.ToJSON y volver a comprobarlo más tarde contra los campos actuales.

Aplicar y Undo

Un diseño validado sustituye al diseño del motor en un único recálculo: las filas, las columnas y los campos de datos en orden, y después los filtros, las ordenaciones y Top N sobre el primer campo con ese nombre en las filas, o si no en las columnas. Undo restaura el diseño anterior a la última respuesta aplicada, y LastLayout queda entonces vacío.

Preguntas asíncronas

Un proveedor puede tardar varios segundos en responder. AskAsync ejecuta solo la petición en un hilo de trabajo, de modo que la ventana sigue respondiendo. OnBeforeAsk se dispara en el hilo principal antes de que empiece la petición, y la respuesta se valida y se aplica en el hilo principal mediante Synchronize, así que OnLayout, OnApplied y OnError se disparan también allí.

- Una pregunta cada vez: AskAsync devuelve False mientras Busy es True, y OnError lo notifica.
- No use el chat desde su propio código mientras Busy sea True.
- Liberar la consulta mientras se ejecuta una petición espera al hilo de trabajo y descarta la respuesta.

```
uses
  SysUtils, Classes, StdCtrls,
  sgcPivotGridWS_AI;

type
  TAskController = class
  private
    FQuery: TsgcPivotGridAIQuery;
    FAskButton: TButton;
    FStatus: TLabel;
    procedure QueryBeforeAsk(Sender: TObject; var aPrompt: string;
      var aAllow: Boolean);
    procedure QueryApplied(Sender: TObject);
    procedure QueryError(Sender: TObject; const aError: string);
  public
    constructor Create(aQuery: TsgcPivotGridAIQuery; aAskButton: TButton;
      aStatus: TLabel);
    procedure AskClick(const aQuestion: string);
  end;

constructor TAskController.Create(aQuery: TsgcPivotGridAIQuery;
  aAskButton: TButton; aStatus: TLabel);
begin
  inherited Create;
  FQuery := aQuery;
  FAskButton := aAskButton;
  FStatus := aStatus;
  FQuery.OnBeforeAsk := QueryBeforeAsk;
  FQuery.OnApplied := QueryApplied;
```

```
  FQuery.OnError := QueryError;
end;

procedure TAskController.AskClick(const aQuestion: string);
begin
  if FQuery.Busy then
    Exit;
  FAskButton.Enabled := False;
  FStatus.Caption := 'Asking the provider...';
  // the request runs in a worker thread, the window stays responsive
  if not FQuery.AskAsync(aQuestion) then
    FAskButton.Enabled := True;
end;

procedure TAskController.QueryBeforeAsk(Sender: TObject; var aPrompt: string;
  var aAllow: Boolean);
begin
  // main thread, before the request: edit the user message or veto it
  aPrompt := aPrompt + sLineBreak + 'Prefer the year grouping for dates.';
end;

procedure TAskController.QueryApplied(Sender: TObject);
begin
  // main thread: the answer was validated and applied in Synchronize
  FAskButton.Enabled := True;
  FStatus.Caption := FQuery.LastExplanation;
end;

procedure TAskController.QueryError(Sender: TObject; const aError: string);
begin
  FAskButton.Enabled := True;
  FStatus.Caption := aError;
end;
```

Edición y requisitos

Requisito	Detalles
Edición	Solo Registered. El puente sgcPivotGridWS no forma parte de la edición Community.
sgcWebSockets	El puente habla con el proveedor a través de TsgcAIChat de sgcWebSockets (unidad sgcAI), por lo que debe estar instalado sgcWebSockets con sus componentes de IA. Los proveedores son los de TsgcAIChat: OpenAI, Anthropic, Gemini, DeepSeek, Ollama, Grok y Mistral.
Plataformas	Solo Windows, Win32 y Win64, Delphi 7 a 13 y las versiones equivalentes de C++Builder.
Paquetes	sgcPivotGridWS<V> y dclsgcPivotGridWS<V> para Delphi, sgcPivotGridWSC<V> y dclsgcPivotGridWSC<V> para C++Builder. Requieren sgcPivotGrid, sgcUICore y sgcWebSockets.
Clave de API	Cada proveedor necesita su propia clave, asignada en ChatOptions.ApiKey del chat. Ollama se ejecuta en local y no necesita ninguna. No guarde la clave en un archivo de formulario.

TsgcAIChat solo existe cuando sgcWebSockets se compila con `SGC_AI`. Los paquetes que construye el instalador de sgcWebSockets lo definen. Si compila usted mismo las fuentes de sgcWebSockets y el compilador informa de TsgcAI_Chat como identificador no declarado en sgcPivotGridWS_AI, añada `SGC_CUSTOM_INDY` a las definiciones condicionales de esa compilación: `SGC_AI` depende de él.

La demo AskYourPivot

`Demos\PivotGrid\Delphi\AskYourPivot` coloca un TsgcVCLPivotGrid con 100.000 filas de ventas generadas bajo un cuadro de pregunta. Elija el proveedor, compruebe el modelo y escriba su clave de API, que solo se guarda en memoria. Ask usa AskAsync, Undo restaura el diseño anterior, Examples contiene cuatro preguntas de ejemplo y Show prompt abre el mensaje de sistema, el mensaje de usuario y el esquema. Ejecútela con `--test` para comprobarla con respuestas predefinidas, sin red y sin clave. Consulte Demos.

Véase también

- TsgcPivotGridAIQuery
- Áreas
- Filtros
- Top N

Descripción general de .NET

sgcPivotGrid .NET es un port administrado en C# de la biblioteca Delphi sgcPivotGrid, no un contenedor alrededor de ella. Las clases están en el espacio de nombres **esegece.sgcPivotGrid** y conservan los nombres de Delphi, de modo que **TsgcPivotEngine**, **TsgcPivotField**, **TsgcPivotResult**, **TsgcPivotGridRenderer**, **TsgcPivotFormatRule**, **TsgcPivotSource_DataSet** y **TsgcPivotSource_Table** se leen igual en ambos lenguajes. Las vistas siguen el patrón de nombres de Delphi: **TsgcWinFormsPivotGrid** y **TsgcWinFormsPivotFieldList** para WinForms, **TsgcWpfPivotGrid** y **TsgcWpfPivotFieldList** para WPF, y **TsgcAvaloniaPivotGrid** y **TsgcAvaloniaPivotFieldList** para Avalonia. Todas dibujan con el port .NET de **sgcUICore**, el mismo lienzo, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, de modo que un pivot se ve igual en cada host y en la salida PNG y SVG sin interfaz.
El motor y su aritmética son los de Delphi, operación por operación: el almacén de columnas tipado, las ocho agregaciones, la agrupación de fechas, los subtotales y totales generales, los filtros de etiqueta y de valor, la ordenación, expandir y contraer, el drill-down a las filas de origen, las reglas de formato condicional y la exportación a CSV, JSON y HTML. La salida SVG del renderizador es idéntica byte a byte a la de Delphi para la misma escena, de modo que un informe generado en un servidor Linux coincide con el que dibuja una aplicación de escritorio Delphi.

Paquetes

El port se divide en paquetes NuGet, de modo que un proyecto solo incorpora lo que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcPivotGrid	El motor (**TsgcPivotEngine**) con su almacén de columnas, campos, agregaciones, agrupación de fechas, totales, filtros, ordenación, drill-down e instantánea (**TsgcPivotResult**), las reglas de formato, el renderizador **TsgcPivotGridRenderer** con la salida SVG, el modelo de la lista de campos, los dos orígenes y la exportación a CSV, JSON y HTML. En la edición Registered también los campos calculados, mostrar valores como, Top N y el JSON de diseño. Sin dependencias nativas.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPivotGrid.Rendering	La salida PNG (**SaveToPNG**), mediante SkiaSharp.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPivotGrid.WinForms	**TsgcWinFormsPivotGrid**, **TsgcWinFormsPivotFieldList** y su ventana emergente de miembros.	net472, net8.0-windows, net9.0-windows
esegece.sgcPivotGrid.Wpf	**TsgcWpfPivotGrid**, **TsgcWpfPivotFieldList** y su ventana emergente de miembros.	net472, net8.0-windows, net9.0-windows
esegece.sgcPivotGrid.Avalonia	**TsgcAvaloniaPivotGrid**, **TsgcAvaloniaPivotFieldList** y su ventana emergente de miembros, en Avalonia 11.3.	net8.0, net9.0

Cada paquete de host depende de esegece.sgcPivotGrid.Rendering y del paquete sgcUICore del mismo host, por lo que basta con añadir el paquete del host.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcPivotGrid, .Rendering, .WinForms, .Wpf y .Avalonia	Todo: las tres cuadrículas y listas de campos, el motor, ambos orígenes, todas las agregaciones, la agrupación de fechas, los totales, los filtros, la ordenación, el drill-down, las reglas de formato, la exportación, la salida sin interfaz y las cuatro funciones avanzadas: campos calculados, mostrar valores como, Top N con Others y persistencia del diseño.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcPivotGrid.WinForms.Community.	Todo excepto las cuatro funciones avanzadas: **CalculatedFields**, las propiedades **ShowAs**, **ShowAsAxis** y **ShowAsBaseItem**, **TopN**, **TopNOthers** y **OthersCaption**, los métodos **SaveLayout** y **LoadLayout**, y **GetExpandState** y **SetExpandState**.

Es la misma división que en las ediciones de Delphi, véase Ediciones.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	TsgcVCLPivotGrid y TsgcVCLPivotFieldList	Windows, en .NET Framework 4.7.2 y .NET 8 y 9.
WPF	TsgcFMXPivotGrid y TsgcFMXPivotFieldList	Windows, en .NET Framework 4.7.2 y .NET 8 y 9.
Avalonia	TsgcFMXPivotGrid y TsgcFMXPivotFieldList	Windows, Linux y macOS, en .NET 8 y 9.
Sin interfaz	TsgcPivotGridRenderer	Cualquier plataforma de .NET Standard 2.0, incluidos los servidores Linux.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Enlace de datos en .NET
- Renderizado sin interfaz en .NET
- Demos de .NET

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host; el núcleo y el paquete de renderizado se instalan con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcPivotGrid.WinForms
dotnet add package esegece.sgcPivotGrid.Wpf
dotnet add package esegece.sgcPivotGrid.Avalonia
```

Un servidor o una aplicación de consola sin ventana añade el núcleo y, para PNG, el paquete de renderizado:

```
dotnet add package esegece.sgcPivotGrid
dotnet add package esegece.sgcPivotGrid.Rendering
```

Edición Community

```
dotnet add package esegece.sgcPivotGrid.WinForms.Community
dotnet add package esegece.sgcPivotGrid.Wpf.Community
dotnet add package esegece.sgcPivotGrid.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcPivotGrid.WinForms" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los Community, nunca ambos: contienen los mismos tipos en el mismo espacio de nombres. El código que usa una función avanzada no compila contra los paquetes Community; protéjalo con `#if !SGC_COMMUNITY` cuando un proyecto compila ambas ediciones, como hace el ejemplo de Diferencias de .NET respecto a Delphi.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto de .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto de .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:pivot="clr-namespace:esegece.sgcPivotGrid;assembly=esegece.sgcPivotGrid.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Asigne el espacio de nombres en XAML con `xmlns:pivot="using:esegece.sgcPivotGrid"`.
- Sin interfaz: esegece.sgcPivotGrid por sí solo basta para el motor, ambos orígenes, CSV, JSON, HTML y SVG. Añada esegece.sgcPivotGrid.Rendering para PNG; incorpora SkiaSharp y sus recursos nativos.

En C# cada host necesita `using esegece.sgcPivotGrid;`, y `using esegece.sgcUICore;` para **TsgcUITheme**, **sgcUICore_Theme** y los demás tipos de los temas.

Véase también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Renderizado sin interfaz en .NET
- Demos de .NET

Inicio rápido WinForms

Este ejemplo construye un formulario WinForms con una cuadrícula pivot y su lista de campos sobre 50.000 filas de ventas generadas: Region y Product en las filas, los años de la columna Date en las columnas y la suma de las ventas en las celdas. Una casilla cambia la cuadrícula y la lista de campos entre el tema claro y el oscuro, y un doble clic en una celda lee las filas de origen que hay detrás.

Pasos

- Cree un proyecto WinForms `net8.0-windows` y añada el paquete **esegece.sgcPivotGrid.WinForms**, véase Instalación desde NuGet.
- Cree un **TsgcWinFormsPivotGrid** y acóplelo. Su propiedad **Pivot** es el **TsgcPivotEngine** que contiene las filas y el diseño.
- Cargue las filas entre **BeginLoad** y **EndLoad**: **DefineColumn** nombra cada columna y fija su tipo, **AppendRow** añade los valores en el orden de las columnas. El motor recalcula una sola vez, en el EndLoad más externo.
- Construya el diseño entre **BeginUpdate** y **EndUpdate**: **AddField** coloca una columna en las filas, las columnas o los filtros, con una agrupación de fechas opcional, y **AddDataField** añade una medida con su agregación.
- Cree una **TsgcWinFormsPivotFieldList** y asigne su **PivotGrid** a la cuadrícula. El usuario arrastra los campos entre las cuatro áreas y filtra los miembros.

Ejemplo

Intro o un doble clic en una celda dispara **OnDrillDown**; **DrillDown** del motor devuelve las filas de origen de la celda como una tabla que puede mostrar una cuadrícula sgcGrid o un segundo pivot, véase Enlace de datos en .NET. Un clic en el botón +/- de un miembro, o un doble clic en el encabezado de una fila, lo expande y lo contrae.

Véase también

- Instalación desde NuGet
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- TsgcVCLPivotGrid
- Drill-down
- Demos de .NET

Inicio rápido WPF

El mismo pivot en WPF, construido en código: un **DockPanel** con una casilla de tema arriba, la lista de campos a la derecha y la cuadrícula en el resto de la ventana. La columna Date va dos veces a las columnas, agrupada por año y por trimestre, y el área de valores contiene dos medidas de la misma columna, la suma y el promedio de las ventas. Un clic en una celda muestra su texto en la barra de título.

Pasos

- Cree un proyecto WPF `net8.0-windows` y añada el paquete **esegece.sgcPivotGrid.Wpf**.
- Cree un **TsgcWpfPivotGrid** y una **TsgcWpfPivotFieldList**, añada ambos a la ventana y asigne **PivotGrid** de la lista de campos a la cuadrícula.
- Rellene **Pivot** como en el ejemplo de WinForms. El motor, el diseño y los eventos son los mismos en todos los hosts.
- Lea el resultado entre **Lock** y **Unlock** del motor. **Result** es una instantánea inmutable, de modo que un recálculo nunca cambia las celdas que está leyendo.

Ejemplo

La cuadrícula WPF dibuja sus propias barras de desplazamiento, como la cuadrícula FireMonkey, y trabaja en píxeles independientes del dispositivo: **HitTest** recibe DIP. Las opciones de visualización son propiedades CLR simples, asígnelas en código o como atributos XAML.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- TsgcFMXPivotGrid

Inicio rápido Avalonia

El mismo pivot en Avalonia, para Windows, Linux y macOS: las cantidades de cuatro productos por región. Una casilla está enlazada a **AlternateRows** de la cuadrícula, y el título sigue a la fila seleccionada.

Pasos

- Cree un proyecto de escritorio Avalonia `net8.0` con **Avalonia.Desktop** y **Avalonia.Themes.Fluent** 11.3, y añada el paquete **esegece.sgcPivotGrid.Avalonia**.
- Añada el **FluentTheme** a los estilos de la aplicación: la ventana emergente de miembros de la lista de campos usa las plantillas Fluent de las casillas y del cuadro de búsqueda.
- Cree un **TsgcAvaloniaPivotGrid** y una **TsgcAvaloniaPivotFieldList**, asigne **PivotGrid** de la lista de campos a la cuadrícula y rellene **Pivot** como en los demás hosts.

Ejemplo

Cada opción de visualización de la cuadrícula Avalonia es una propiedad con estilo, por lo que admite enlaces, estilos y valores XAML, y **SelectedRow** y **SelectedCol** son propiedades directas bidireccionales. En macOS la tecla Cmd funciona como Ctrl para Ctrl+Inicio y Ctrl+Fin, y una rueda horizontal desplaza las columnas.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Enlace de datos en .NET
- Diferencias de .NET respecto a Delphi
- TsgcFMXPivotGrid

Clases de .NET

Todas las clases están en el espacio de nombres **esegece.sgcPivotGrid**. Las cuadrículas y las listas de campos son hosts delgados sobre el motor, el renderizador y el modelo de lista de campos independientes del framework, que comparten los hosts, la salida sin interfaz y los orígenes.

Clases de host

Host	Clase	Clase base	Delphi
WinForms	TsgcWinFormsPivotGrid	TsgcUIWinFormsControl	TsgcVCLPivotGrid
WinForms	TsgcWinFormsPivotFieldList	TsgcUIWinFormsControl	TsgcVCLPivotFieldList
WPF	TsgcWpfPivotGrid	TsgcUIWpfControl	TsgcFMXPivotGrid
WPF	TsgcWpfPivotFieldList	TsgcUIWpfControl	TsgcFMXPivotFieldList
Avalonia	TsgcAvaloniaPivotGrid	TsgcUIAvaloniaControl	TsgcFMXPivotGrid
Avalonia	TsgcAvaloniaPivotFieldList	TsgcUIAvaloniaControl	TsgcFMXPivotFieldList

La lista de verificación de miembros de cada lista de campos es **TsgcWinFormsPivotMemberPopup** (un Form), **TsgcWpfPivotMemberPopup** (una Window) o **TsgcAvaloniaPivotMemberPopup** (un Popup). Las tres cuadrículas implementan **IsgcPivotEngineHost**, para que un origen o una lista de campos encuentre su motor, e **IsgcUITable**, para que un gráfico de sgcCharts u otro pivot pueda leer su resultado.

Miembros de la cuadrícula

Las tres cuadrículas comparten los mismos miembros.

Kind	Miembros
Motor y dibujo	**Pivot** (el **TsgcPivotEngine**), **Renderer**, **FormatRules**.
Opciones de visualización	**RowLayout**, **RowHeight**, **HeaderHeight**, **IndentWidth**, **MinColumnWidth**, **MaxColumnWidth**, **AutoColumnWidths**, **GridLines**, **AlternateRows**, **TotalsBold**, **ShowExpandButtons**, **Theme**, **AccentColor**, **FontSize**.
Posición	**TopRow**, **ScrollX**, **SelectedRow**, **SelectedCol**, **ScrollToCell**, **HitTest**.
Salida	**Recalculate**, **SaveToPNG** y **SaveToSVG** (un archivo o un flujo, con ancho y alto opcionales; 0 es la vista completa), **SaveToCSV**, **SaveToJSON**, **SaveToHTML**. Edición Registered: **SaveLayoutToString**, **LoadLayoutFromString**, **SaveLayoutToFile**, **LoadLayoutFromFile**.
Eventos	**OnCellClick**, **OnCellDblClick**, **OnDrillDown**, **OnExpandChange**, **OnGetCellStyle**, **OnGetCellText**, **OnSelectionChange**, **OnColumnResized**.

Las listas de campos tienen **PivotGrid** (cualquier **IsgcPivotEngineHost**), **Engine**, **Theme**, **AccentColor**, **FontSize**, **ShowFilterArea**, **DeferUpdate** con **ApplyChanges**, y los eventos **OnBeforeFieldMove** y **OnFieldMoved**.

Tipos independientes del framework

Tipo	Descripción
TsgcPivotEngine	El almacén de columnas, el diseño, el cálculo, la instantánea, el drill-down y la exportación: **BeginLoad**, **DefineColumn**, **AppendRow**, **EndLoad**, **LoadFromTable**, **AddField**, **AddDataField**, **MoveField**, **RemoveField**, **ClearLayout**, **ExpandAll**, **CollapseAll**, **Lock**, **Unlock**, **Result**, **ResultTable**, **GetSourceRows**, **DrillDown**, **GetMemberLabels** y los eventos **OnCalculated**, **OnLayoutChange** y **OnFilterRow**. Véase TsgcPivotEngine.
TsgcPivotFields, TsgcPivotField	Un campo del diseño: **FieldName**, **Caption**, **Area**, **AreaIndex**, **DateGrouping**, **Aggregation**, **DisplayFormat**, **SortBy**, **SortOrder**, **SortMeasure**, **Subtotals**, **Expanded**, **FilterValues**, **FilterExclude**, **LabelFilter**, **ValueFilter** y, en la edición Registered, **TopN**, **TopNOthers**, **ShowAs**, **ShowAsAxis** y **ShowAsBaseItem**. Véase TsgcPivotField.
TsgcPivotCalculatedFields, TsgcPivotCalculatedField	Columnas numéricas calculadas por fila de origen a partir de una expresión. Edición Registered. Véase TsgcPivotCalculatedField.
TsgcPivotResult, TsgcPivotItem	La instantánea inmutable de un recálculo: **ItemCount**, **Item**, **NodeLabel**, **NodePath**, **LevelCount**, **MeasureCount**, **DataColCount**, **CellText**, **CellValue** y **RawValue**. Un **TsgcPivotItem** es un struct con **Kind** (`pvkMember` , `pvkTotal` , `pvkGrandTotal`), **Level**, **Node**, **Expandable** y **Expanded**.
TsgcPivotResultTable, TsgcPivotSourceView	El resultado de un motor y las filas de origen de un drill-down, ambos como **IsgcUITable**.
TsgcPivotSource_DataSet	Carga una `System.Data.DataTable` y la recarga cuando la tabla cambia. Véase Enlace de datos en .NET.
TsgcPivotSource_Table	Carga cualquier **IsgcUITable** y la recarga cuando cambia su versión. Véase TsgcPivotSource_Table.
TsgcPivotGridRenderer, TsgcPivotRenderOptions, TsgcPivotRenderState	Dibuja una instantánea en un **TsgcUICanvas**, realiza las pruebas de impacto y la escribe como SVG o, con esegece.sgcPivotGrid.Rendering, como PNG. Véase Renderizado sin interfaz en .NET.
TsgcPivotFormatRules, TsgcPivotFormatRule	Formato condicional: reglas de valor y escalas de color. Véase TsgcPivotFormatRule.
TsgcPivotFieldListModel	La lista de campos sin ventana: las áreas, los chips, arrastrar y soltar y el estado de los filtros, dibujados por las tres listas de campos.
sgcPivotGrid_IO	Exportación estática a CSV, JSON y HTML de un resultado, con los argumentos predeterminados de Delphi. El motor y las cuadrículas la envuelven en **SaveToCSV**, **SaveToJSON** y **SaveToHTML**.

Enumeraciones

Tipo	Valores
TsgcPivotArea	`pvaRow` , `pvaColumn` , `pvaData` , `pvaFilter`
TsgcPivotAggregation	`pvfSum` , `pvfCount` , `pvfAverage` , `pvfMin` , `pvfMax` , `pvfCountDistinct` , `pvfStdDev` , `pvfVar`
TsgcPivotDateGrouping	`pvdNone` , `pvdYear` , `pvdQuarter` , `pvdMonth` , `pvdWeek` , `pvdDay`

TsgcPivotDataType	`pvtText`, `pvtInteger`, `pvtFloat`, `pvtDateTime`, `pvtBoolean`
TsgcPivotAxisKind	`pvxRows`, `pvxColumns`
TsgcPivotSortBy, TsgcPivotSortOrder	`pvsLabel`, `pvsValue`; `pvoAscending`, `pvoDescending`
TsgcPivotRowLayout	`pvlCompact`, `pvlTabular`
TsgcPivotShowAs	`pvwValue`, `pvwPercentOfRow`, `pvwPercentOfColumn`, `pvwPercentOfGrandTotal`, `pvwRunningTotal`, `pvwDifferenceFrom`, `pvwRank`. Edición Registered.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Enlace de datos en .NET
- Renderizado sin interfaz en .NET
- Arquitectura

Diferencias de .NET respecto a Delphi

El port .NET conserva los nombres, los valores predeterminados, la aritmética y las reglas de renderizado de Delphi. Su salida SVG y la exportación a CSV, JSON y HTML de datos ordinarios son idénticas byte a byte a las de Delphi. Las diferencias que siguen proceden de los frameworks de los hosts y de la biblioteca de clases de .NET.

Sin streaming de componentes

En .NET no hay DFM, por lo que no hay editores de colecciones en tiempo de diseño, ni lista desplegable de FieldName, ni los verbos **Recalculate** y **Clear Layout**. Construya el diseño en código con **AddField** y **AddDataField**, entre **BeginUpdate** y **EndUpdate**, para que el motor recalcule una sola vez. El motor de Delphi aplaza el trabajo mientras un formulario se carga; en .NET eso lo hacen **BeginUpdate** y **BeginLoad**. Para conservar un diseño entre sesiones use el JSON de diseño de la edición Registered, véase Persistencia del diseño.

Propiedades indexadas, eventos y listas

- Una propiedad indexada de Delphi es un método: `Result.Item(axis, i)`, `NodeLabel(axis, node)`, `SourceColumnName(i)` y, en las ventanas emergentes de miembros, `Members(i)` y `MemberChecked(i)` con `SetMemberChecked(i, value)`. Una función de Delphi sigue siendo un método: `SourceRowCount()`, `MeasureCount()`, `DataColCount()`. Las colecciones conservan su indexador, como `Fields[0]`.
- Los eventos son propiedades delegado con los nombres de Delphi (**OnCalculated**, **OnDrillDown**...). Asigne un manejador con `=` o encadene varios con `+=`. Un parámetro `var` de Delphi es un parámetro `ref` (`aHandled`, `aAccept`, `aAllow`), y una lambda con un parámetro `ref` declara el tipo de todos los parámetros. **TNotifyEvent** es **TsgcUINotifyEvent**.
- Un parámetro **TStrings** es una `IList<string>`. **FilterValues** es un **TsgcPivotFilterValues**, una lista ordenada de cadenas cuyo **IndexOf** ignora las mayúsculas, como TStringList. El filtro en sí compara las etiquetas de los miembros distinguiendo mayúsculas.
- **Assign** de un campo, un filtro, una regla o un campo calculado desde otra clase lanza `InvalidCastException`, el EConvertError de Delphi.

Valores, fechas y culturas

AppendRow recibe valores CLR: null y DBNull son celdas vacías, los tipos enteros son `pvtInteger`, `float`, `double` y `decimal` son `pvtFloat`, `DateTime` es `pvtDateTime`, `bool` es `pvtBoolean`, y `string` y `char` son texto. Las fechas se guardan como fechas OLE, el TDateTime de Delphi, de modo que la agrupación de fechas y los números de semana coinciden con Delphi. El texto se analiza solo con la cultura invariante, mientras que Delphi prueba también la configuración regional del usuario, y las etiquetas de los miembros, **CellText** y las exportaciones formatean números y fechas con valores invariantes (punto como separador decimal, fechas como MM/dd/yyyy), nunca con la cultura del hilo.

DataTable en lugar de TDataSet

TsgcPivotSource_DataSet enlaza una `System.Data.DataTable`, el equivalente .NET de un TDataSet de Delphi. Una DataTable siempre está abierta y no tiene cursor, por lo que no hay registro actual, ni marcador, ni DisableControls. Los eventos `RowChanged`, `RowDeleted` y `TableCleared` ocupan el lugar del data link de Delphi; las ediciones sin terminar y las filas nuevas separadas se ignoran, igual que Delphi ignora las ediciones e inserciones sin confirmar. Véase Enlace de datos en .NET.

Sin notificación de liberación

Cuando **PivotGrid** de un origen o de una lista de campos es un `IComponent`, como la cuadrícula WinForms, el origen escucha su evento `Disposed` y entonces borra **PivotGrid**. Las cuadrículas WPF y Avalonia no son componentes: asigne null a **PivotGrid** antes de descartar la cuadrícula. **Engine** nunca se vigila; bórrelo antes de que el mo-

tor desaparezca. **TsgcPivotSource_DataSet** y **TsgcPivotSource_Table** son `IDisposable`; **Dispose** retira sus oyentes de la tabla.

Hilos

El cálculo es síncrono, como en Delphi. Un recálculo se ejecuta en el hilo que lo dispara y publica allí la nueva instantánea; las cuadrículas se repintan entonces en su hilo de interfaz, mediante un mensaje de ventana enviado en WinForms, `Dispatcher.BeginInvoke` en WPF y `Dispatcher.UIThread.Post` en Avalonia. El código que lee **Result** desde otro hilo mantiene **Lock** y **Unlock** del motor.

Barras de desplazamiento, unidades y propiedades por host

- **TsgcWinFormsPivotGrid** usa las barras de desplazamiento nativas de Windows y píxeles, como la cuadrícula VCL. **TsgcWpfPivotGrid** y **TsgcAvaloniaPivotGrid** dibujan sus propias barras de desplazamiento con arrastre del control deslizante y paginación en la pista, como la cuadrícula FireMonkey, y trabajan en píxeles independientes del dispositivo.
- El **BorderStyle** de Delphi es **BorderThickness** (1 por defecto) y **BorderBrush** en WPF y Avalonia. **TsgcWpfPivotGrid** declara su propio **FontSize** con `new`, y **TsgcAvaloniaPivotGrid** su propio **Theme** de tipo **TsgcUITheme**, que oculta el tema de control de Avalonia. A través de una referencia de tipo `Control` son las propiedades del framework, que la cuadrícula no lee.
- Las opciones de WPF son propiedades CLR. Las opciones de Avalonia son propiedades con estilo que admiten enlaces y estilos, véase Enlace de datos en .NET.
- La ventana emergente de miembros de la lista de campos es un Form en WinForms, una Window sin borde propiedad de la ventana de la lista de campos en WPF, y un Popup que se cierra al hacer clic fuera en Avalonia. En macOS la tecla Cmd funciona como Ctrl.

Véase también

- Descripción general de .NET
- Clases de .NET
- Enlace de datos en .NET
- Inicio rápido WPF
- Inicio rápido Avalonia
- TsgcPivotEngine

Enlace de datos en .NET

El motor mantiene su propia copia tipada de las filas de origen, de modo que un pivot nunca lee la fuente de datos mientras calcula o dibuja. Hay cuatro formas de rellenarlo:

Forma	Úsela para
BeginLoad, **DefineColumn**, **AppendRow**, **EndLoad**	Filas que genera su código o lee de cualquier lector. Todos los inicios rápidos la usan.
TsgcPivotSource_DataSet	Una `System.Data.DataTable`, recargada cuando cambian sus filas.
TsgcPivotSource_Table	Cualquier **IsgcUITable**: las cuadrículas y el almacén de filas de sgcGrid .NET, otro motor pivot o su resultado, recargado cuando cambia la versión de la tabla.
LoadFromTable	Una copia única de una **IsgcUITable**, como las filas de origen de un drill-down.

Un origen rellena el motor de una cuadrícula cuando se asigna **PivotGrid**, o un motor sin interfaz cuando se asigna **Engine**; **Engine** prevalece si se asignan ambos. Cada carga reemplaza las filas en un único bloque de carga, de modo que el motor recalcula una vez y conserva su diseño.

DataTable

El ejemplo imprime `4 source rows, 3 columns`: la columna Notes no está en **Columns**, y la fila añadida después del enlace recargó el motor. El tipo de una columna de DataTable fija el tipo de la columna del pivot:

Tipo de columna de DataTable	Tipo de columna del pivot
`byte` , `short` , `int` , `long` y los demás tipos enteros	`pvtInteger`
`float` , `double` , `decimal`	`pvtFloat`
`DateTime` , `DateTimeOffset` , `TimeSpan`	`pvtDateTime`
`bool`	`pvtBoolean`
Columnas binarias (`byte[]` y los demás arreglos excepto `char[]`)	Nunca se cargan.
Cualquier otro	`pvtText` , con la cultura invariante.

Con **LiveUpdates** un cambio confirmado recarga el motor: una fila añadida, modificada, revertida o eliminada, y una tabla vaciada. Una fila en modo de edición carga sus valores actuales, de modo que una carga nunca confirma una edición pendiente, y una fila nueva se ignora hasta que se añade a la tabla. **Load** carga en cualquier momento y **Refresh** recarga cuando **Active** es true.

IsgcUITable

PivotOfCell construye un segundo pivot sobre las filas que hay detrás de una celda, para una ventana de detalle. **Follow** construye un pivot de un pivot: un motor es una **IsgcUITable** de su resultado, con una columna por campo de fila y una por columna de datos, con el nombre de los miembros de columna y de la medida, por ejemplo `Online | Sum of Sales`. Cada recálculo del primer motor cambia la versión de la tabla y recarga el segundo. La misma pro-

piedad **Table** admite una cuadrícula sgcGrid .NET, cuyas filas edita un usuario, como hace la demo GridToPivotToChart.

Enlazar las opciones de visualización

En Avalonia cada opción de visualización de la cuadrícula (**RowLayout**, **RowHeight**, **HeaderHeight**, **IndentWidth**, **MinColumnWidth**, **MaxColumnWidth**, **AutoColumnWidths**, **GridLines**, **AlternateRows**, **TotalsBold**, **ShowExpandButtons**, **Theme**, **AccentColor**, **FontSize**, **BorderThickness**, **BorderBrush**) es una propiedad con estilo, y **SelectedRow** y **SelectedCol** son propiedades directas bidireccionales. Admiten enlaces, estilos y valores XAML, véase Inicio rápido Avalonia. Un cambio hecho en código en **Renderer.Options** se copia de vuelta a las propiedades. La lista de campos de Avalonia también tiene **Theme**, **AccentColor**, **FontSize**, **ShowFilterArea** y **DeferUpdate** como propiedades con estilo.
En WPF y WinForms las opciones son propiedades CLR: asígnelas en código, o como atributos en XAML en WPF.

Véase también

- Descripción general de .NET
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- TsgcPivotSource_DataSet
- TsgcPivotSource_Table
- Interoperabilidad con gráficos

Renderizado sin interfaz en .NET

El motor y el renderizador no necesitan ventana, de modo que un servicio, una aplicación web o un trabajo de consola en Windows, Linux o macOS puede construir un pivot y escribirlo como PNG, SVG, CSV, JSON o HTML. Este ejemplo carga 10.000 filas de ventas, conserva el canal online con un campo de filtro, imprime el resultado, lee las filas de origen que hay detrás de una celda y escribe los cinco archivos.

Ejemplo

La salida de la consola:

```
Region                2024          2025   Grand Total
East            145,862.61    155,105.09    300,967.70
North           162,608.24    155,080.86    317,689.10
South           154,901.83    155,794.10    310,695.93
West            156,547.53    157,665.06    314,212.59
Grand Total     619,920.21    623,645.11  1,243,565.32
The first cell adds up 580 source rows
```

Notas

- Una celda se direcciona mediante un elemento de fila y una columna de datos. Con varias medidas, las columnas de datos de un elemento de columna están una junto a otra: la columna de datos es el elemento de columna por **MeasureCount** más la medida.
- **TsgcPivotRenderState** contiene la posición de desplazamiento, la selección y los anchos de columna; `sgcPivotGrid_Render.sgcPivotRenderState()` devuelve el estado predeterminado. Un ancho y un alto de 0, el valor predeterminado, dibujan la vista completa en lugar de una pantalla.
- **SaveToSVG** forma parte de esegece.sgcPivotGrid y no tiene dependencias nativas. **SaveToPNG** es un método de extensión de esegece.sgcPivotGrid.Rendering, que dibuja con SkiaSharp.
- Las cuadrículas tienen los mismos métodos: `Grid.SaveToPNG(fileName)` escribe la vista completa de la cuadrícula con sus opciones y reglas de formato.
- La exportación a CSV escribe los valores sin procesar con precisión completa; la exportación a HTML escribe los textos de celda con formato.

Véase también

- Descripción general de .NET
- Clases de .NET
- TsgcPivotGridRenderer
- Exportación
- Inicio rápido sin interfaz
- Demos de .NET

Demos de .NET

Las demos están en `.net\demos\PivotGrid`, una carpeta por demo con su proyecto, su `Program.cs` y un README.md. Cada demo tiene un conmutador `--test` que ejecuta una autocomprobación fuera de pantalla, escribe líneas PASS y FAIL y termina con el número de comprobaciones fallidas, como el conmutador `/test` de las demos de Delphi.

Demo	Host	Muestra
SalesPivot	WinForms	100.000 filas de ventas generadas con una lista de campos: Region y Country en las filas, la columna Date como Year, Quarter y Month en las columnas, el canal como filtro, ordenación por valor, una escala de color, expandir y contraer, y exportación a CSV, HTML y PNG.
SalesPivotWpf	WPF	El mismo pivot en **TsgcWpfPivotGrid**.
SalesPivotAvalonia	Avalonia	El mismo pivot en **TsgcAvaloniaPivotGrid**, para Windows, Linux y macOS.
GridToPivotToChart	WinForms	Una cuadrícula sgcGrid .NET, editable en la edición Registered, alimenta el pivot mediante **TsgcPivotSource_Table**, y el resultado del pivot alimenta un gráfico sgcCharts .NET mediante **IsgcUITable**.
MillionRows	WinForms	Un millón de filas generadas con tres medidas (la suma, el recuento distinto de 100.000 clientes y la desviación estándar), con el tiempo de la carga y de cada recálculo.
HeadlessReport	Consola	Lee un archivo CSV, o escribe primero un orders.csv de muestra con 5.000 pedidos generados, y escribe el pivot como PNG y SVG con **TsgcPivotGridRenderer**, y como HTML, CSV y JSON con el motor, sin ventana. Funciona en Windows, Linux y macOS.

Compilar las demos

Dentro del árbol del producto las demos se compilan contra los proyectos de la biblioteca. El zip de demos usa por defecto los paquetes NuGet de la edición Community. Ninguna usa una función avanzada, por lo que todas las demos compilan y se ejecutan en ambas ediciones. Compile una demo con los paquetes Registered asignando **SgcPivotGridEdition**:

```
dotnet build SalesPivot\SalesPivot.csproj -p:SgcPivotGridEdition=Registered
dotnet run --project HeadlessReport\HeadlessReport.csproj -- --test
```

GridToPivotToChart restaura además los paquetes sgcGrid y sgcCharts de la misma edición.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Renderizado sin interfaz en .NET
- Demos

Solución de problemas

La cuadrícula permanece vacía

Compruebe que llegaron datos: SourceRowCount del motor es 0 cuando el origen no está Active o no tiene DataSet, Table ni PivotGrid. Compruebe los nombres: un campo de fila, columna o filtro cuyo FieldName no coincide con ninguna columna de origen se ignora, y un campo de datos con un nombre desconocido queda vacío. Los nombres se comparan sin distinguir mayúsculas, y SourceColumnName enumera las columnas cargadas. Con AutoRecalculate False, llame a Recalculate. Un filtro que no deja ninguna fila de origen también da una tabla dinámica vacía.

La tabla dinámica no sigue al conjunto de datos

TsgcPivotSource_DataSet recarga mientras Active y LiveUpdates son True, cuando el conjunto de datos notifica un cambio de datos en estado de exploración: un post, un append, un delete, un refresh, First o Last. Next, Prior y MoveBy no recargan, y una edición recarga solo después de hacer el post.

La carga es lenta

AppendRow fuera de un bloque BeginLoad y EndLoad recalcula después de cada fila. Cargue dentro de un bloque, consulte Rendimiento.

EsgcPivotError: The pivot layout needs more than ... cells

El diseño crea más celdas que MaxCells, normalmente porque un campo con muchos miembros está en las filas y otro en las columnas. Agrupe las fechas, mueva un campo a los filtros, use Top N o aumente MaxCells.

EsgcPivotError: Calculated field ...

La expresión de ese campo calculado tiene un error de sintaxis, o los campos calculados se refieren entre sí en círculo. El mensaje nombra el campo. Un nombre de campo que todavía no está cargado no es un error, el campo calculado permanece vacío hasta que llegue la columna. Consulte Campos calculados.

EsgcPivotError: Invalid pivot layout

LoadLayoutFromString o LoadLayoutFromFile recibió un JSON mal formado, o una raíz que no es un objeto. El diseño se deja sin cambios.

Undeclared identifier CalculatedFields, TopN, ShowAs o SaveLayoutToString

La edición Community omite los campos calculados, mostrar valores como, Top N y la persistencia del diseño, de modo que estos miembros no existen en ella, junto con OthersCaption, TopNOthers, GetExpandState y SetExpandState. Un formulario que guarda estas propiedades no puede cargarse en una compilación Community. Consulte Ediciones.

Las celdas vacías no muestran nada

Establezca EmptyText del motor, por ejemplo a un guion. Las cuadrículas, el renderizador, la exportación HTML y ResultTable lo muestran, y OnGetCellText lo recibe y aún puede sustituirlo. SaveToCSV escribe un campo vacío y SaveToJSON escribe null, de modo que los datos exportados siguen siendo numéricos.

Un total no es la suma de las celdas

Los totales se calculan a partir de las filas de origen: un promedio es el promedio de las filas, y un recuento de distintos cuenta cada valor una vez. Después de un filtro de valor o de Top N, los totales incluyen solo los miembros que quedan. Un campo calculado se evalúa por fila y después se agrega. Consulte Agregaciones y Totales.

Las fechas no se agrupan, o los números se ordenan como texto

DateGrouping solo funciona en una columna de fechas, y una ordenación por etiqueta compara los números por valor solo en una columna numérica. Una columna cuyo primer valor fue una cadena es una columna de texto. Pase valores TDateTime y numéricos a AppendRow, o fije el tipo con DefineColumn(aName, aType) antes de la primera fila.

Una vista de drill-down está vacía, o una violación de acceso al cerrar

Un TsgcPivotSourceView no tiene filas después de que cambien los datos de origen, y lee el almacén de su motor, así que libérelo antes que el motor. Consulte Drill-down.

Valores incorrectos o una excepción cuando otro hilo lee Result

Lea Result solo entre Lock y Unlock del motor. Un recálculo publica una nueva instantánea y libera la anterior.

C++Builder: no matching constructor for TsgcPivotEngine o TsgcPivotGridRenderer

Los encabezados generados no conservan los argumentos por defecto de estos constructores. Pase NULL de forma explícita: `new TsgcPivotEngine(NULL)` y `new TsgcPivotGridRenderer(NULL, NULL)`.

TsgcFMXPivotGrid falta o está vacío

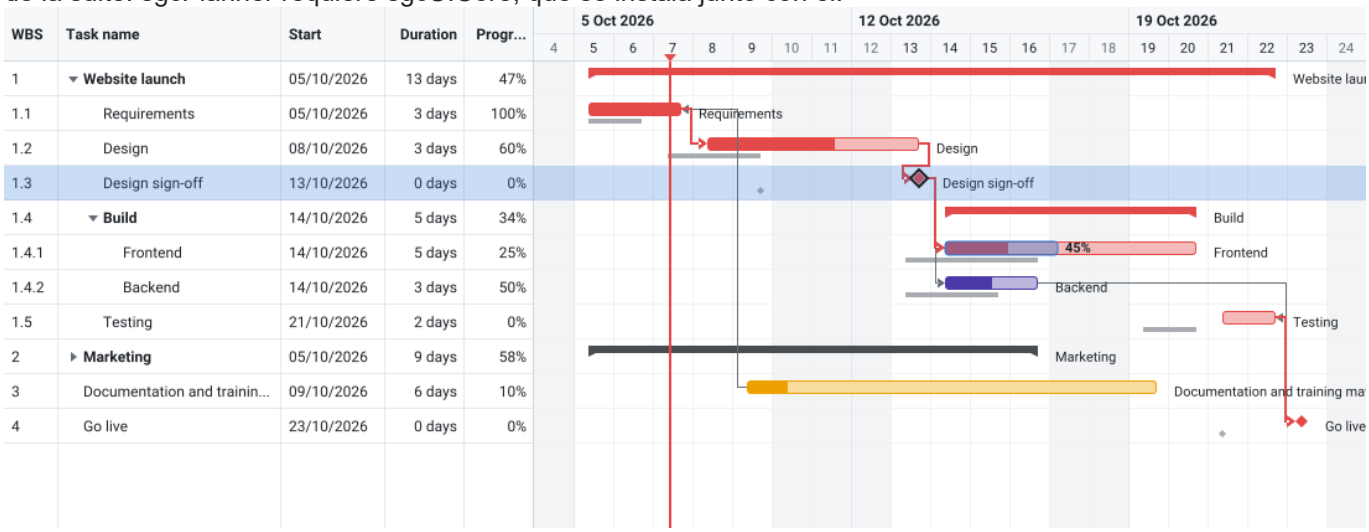
TsgcFMXPivotGrid y TsgcFMXPivotFieldList existen desde Delphi XE6, donde sgcVer.inc define SGC_PIVOTGRID_FMX. Con un compilador más antiguo la unidad sgcPivotGrid_FMX compila sin ellos. En tiempo de diseño están en la página SGC PivotGrid de la paleta mientras hay un formulario de FireMonkey abierto. Consulte Plataformas.

La lista de campos no muestra campos

Establezca PivotGrid de la lista de campos, o Engine para un motor sin cuadrícula. La lista muestra las columnas de origen del motor, por lo que se rellena cuando se cargan los datos.

Introducción

sgcPlanner añade programación y planificación de proyectos a las aplicaciones Delphi y C++Builder, para VCL y FireMonkey. Un objeto de datos, **TsgcPlannerData**, contiene los elementos, los recursos, los estados, las dependencias y el calendario laboral, y tres tipos de vista lo muestran a la vez: **TsgcVCLScheduler** y **TsgcFMXScheduler** (día, semana laboral, semana, mes, agenda y una Timeline de recursos), **TsgcVCLGantt** y **TsgcFMXGantt** (un árbol de tareas con barras, dependencias, programación automática, ruta crítica y líneas base) y **TsgcVCLKanban** y **TsgcFMXKanban** (un tablero con columnas de estado, carriles y límites de WIP). Arrastre una tarjeta a otra columna y el Gantt y el Scheduler muestran al instante el nuevo estado.
Los datos proceden de su código, de un **TDataSet** mediante **TsgcPlannerSource_DataSet**, de un archivo JSON, de un archivo iCalendar (.ics) mediante **TsgcPlannerICalendar** o de un archivo XML de Microsoft Project mediante **TsgcPlannerMSProject**. **TsgcPlannerRenderer** dibuja cualquier vista en PNG y SVG sin ventana, en un servidor o en Linux64.
sgcPlanner forma parte de la suite visual de eSeGeCe. Se dibuja con **sgcUICore**, el paquete de runtime compartido con el canvas, el rasterizador, las fuentes, el escritor de SVG y los temas claro y oscuro, de modo que un planner tiene el mismo aspecto en VCL, en FireMonkey y en la salida sin ventana, y combina con los demás productos de la suite. sgcPlanner requiere sgcUICore, que se instala junto con él.



Ediciones

sgcPlanner se distribuye como Registered, Trial y Community. La edición Community conserva las tres vistas, el arrastrar y soltar, los resúmenes, los hitos, el progreso, el origen DataSet, JSON y la salida sin ventana, y deja fuera la recurrencia, la programación automática con flechas de dependencia, la ruta crítica, las líneas base, la Timeline de recursos, los carriles, los límites de WIP, iCalendar y el XML de MS Project. Consulte Ediciones.

Por dónde empezar

- Arquitectura: el objeto de datos, las notificaciones de cambio, las vistas y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX e Inicio rápido C++Builder: tres vistas sincronizadas en pocos minutos.
- TsgcPlannerData: los elementos, los recursos, los estados, las dependencias y el calendario laboral.
- Las vistas: TsgcVCLScheduler, TsgcVCLGantt, TsgcVCLKanban y sus equivalentes de FireMonkey.
- Los componentes no visuales: TsgcPlannerSource_DataSet, TsgcPlannerICalendar, TsgcPlannerMSProject y TsgcPlannerRenderer.
- Instalación y Plataformas: los paquetes, las carpetas y los IDE y plataformas compatibles.
- Demos: las ocho demos de Delphi y lo que muestra cada una.
- Las características: Datos compartidos, Vistas sincronizadas, Calendario laboral, Recurrencia, Escala de tiempo y zoom, Arrastrar y soltar, Dependencias, Programación automática, Ruta crítica, Líneas base, Carriles y límites de WIP, Vinculación con DataSet, Persistencia JSON, iCalendar, XML de MS Project, Renderizado sin ventana, Temas y Características por edición.
- Solución de problemas: los problemas habituales y sus causas.

Arquitectura

sgcPlanner mantiene los datos separados de las vistas, y las vistas separadas del framework. El objeto de datos, los motores y los modelos de vista usan solo la RTL y sgcUICore. Los controles VCL y FMX son hosts finos alrededor de un modelo de vista, y el renderer sin ventana usa los mismos modelos de vista, de modo que un PNG coincide píxel a píxel con lo que muestra un formulario.

Flujo de datos

1. **Orígenes.** Su código rellena TsgcPlannerData directamente, o TsgcPlannerSource_DataSet lo carga desde TDataSets y escribe de vuelta las ediciones. LoadFromJSON lee un proyecto guardado, TsgcPlannerICalendar importa y exporta archivos .ics y TsgcPlannerMSProject importa y exporta XML de Microsoft Project.
2. **Datos.** TsgcPlannerData es un TComponent con cinco partes: Items, Resources, Statuses, Dependencies y Calendar. Mantiene un índice por Id, la estructura de padres e hijos, la selección y un número de Version que crece con cada cambio.
3. **Motores.** El calendario laboral cuenta minutos laborales, el motor de recurrencia expande las reglas RRULE en ocurrencias y el motor de programación coloca las tareas después de sus predecesoras, calcula la holgura y marca la ruta crítica. Los dos últimos forman parte de la edición Registered.
4. **Modelos de vista.** Cada vista se registra como listener de los datos. Ante un cambio reconstruye solo lo que necesita: un cambio de selección repinta un overlay, un cambio de elemento rehace el diseño en el siguiente fotograma, un cambio de estructura reconstruye las filas, las columnas y los carriles.
5. **Salida.** Los modelos de vista dibujan en un TsgcUICanvas de sgcUICore. Los seis controles los muestran en un formulario con desplazamiento, ratón, teclado y táctil, y TsgcPlannerRenderer los dibuja en PNG y SVG.

Clases

Clase	Unidad	Función
TsgcPlannerData	sgcPlanner_Data	Los datos compartidos: elementos, recursos, estados, dependencias, calendario, selección, notificaciones y JSON.
TsgcPlannerItem	sgcPlanner_Data	Una tarea, tarjeta o evento: fechas, estado, progreso, padre, recursos, etiquetas, recurrencia, restricción y línea base.
TsgcPlannerCalendar	sgcPlanner_Calendar	Los días laborables, el horario de trabajo, los festivos y la aritmética de los minutos laborales.
TsgcPlannerTimeScale	sgcPlanner_TimeScale	El eje de tiempo del Gantt y de la Timeline: zoom, ancho de las marcas, marcas, bandas de tiempo no laboral y ajuste.
TsgcVCLScheduler, TsgcFMXScheduler	sgcPlanner_Scheduler_VCL, sgcPlanner_Scheduler_FMX	Las vistas de calendario: día, semana laboral, semana, mes, agenda y la Timeline de recursos.
TsgcVCLGantt, TsgcFMXGantt	sgcPlanner_Gantt_VCL, sgcPlanner_Gantt_FMX	El árbol de tareas y el diagrama de barras con dependencias, ruta crítica y líneas base.

TsgcVCLKanban, TsgcFMX-Kanban	sgcPlanner_Kanban_VCL, sgcPlanner_Kanban_FMX	El tablero: columnas de estado, carriles, límites de WIP y tarjetas.
TsgcPlannerSource_DataSet	sgcPlanner_Source_DataSet	Carga los datos desde TDataSets, sigue sus cambios y escribe de vuelta las ediciones.
TsgcPlannerICalendar	sgcPlanner_ICal	Importación y exportación de VEVENT de la RFC 5545. Edición Registered.
TsgcPlannerMSProject	sgcPlanner_MSProject	Importación y exportación de XML de Microsoft Project (MSPDI). Edición Registered.
TsgcPlannerRenderer	sgcPlanner_Renderer	Dibuja una vista Scheduler, Gantt o Kanban sin ventana, en un canvas, un PNG o un SVG.
TsgcPlannerScheduleEngine	sgcPlanner_AutoSchedule	Pasada hacia delante y hacia atrás, holgura, ruta crítica, líneas base y detección de ciclos. Edición Registered.

Notificaciones de cambio

- Cada setter de los datos notifica. Envuelva un lote de cambios en BeginUpdate y EndUpdate: los listeners y OnChange se disparan una vez, tras el EndUpdate más externo, con los tipos de cambio (pckItems, pckStructure, pckResources, pckStatuses, pckDependencies, pckCalendar, pckSelection, pckSchedule), el Id del elemento cuando cambió exactamente un elemento y la Version.
- Las vistas son listeners y solo leen los datos. Su manejador de OnChange puede volver a cambiar los datos: los cambios anidados llegan como una ronda más tras la actual, hasta 8 rondas, y entonces se provoca EsgcPlannerError.
- Una edición hecha con el ratón dispara OnItemChanging de los datos una vez por drop, con los campos que están a punto de cambiar, de modo que la aplicación puede rechazarla. La edición se confirma entonces en una sola actualización, de modo que un drop da una notificación a cada vista vinculada.
- Dos o más vistas sobre los mismos datos se mantienen al unísono: con SyncSelection también se comparte la selección, que es SelectedId de los datos.

Tiempo

El modelo guarda sus horas como TsgcPlannerTime, un recuento Int64 de minutos, de modo que la aritmética es exacta y se descartan los segundos. Las propiedades publicadas son TDateTime, y cada una tiene un equivalente TsgcPlannerTime: Start y StartTime, Finish y FinishTime, ProjectStart y ProjectStartTime. sgcPlannerTimeOf y sgcPlannerDateTimeOf de sgcPlanner_DateUtils convierten entre ellos. El Finish de un elemento es exclusivo: un elemento de todo el día el 5 de octubre termina a medianoche del 6 de octubre. Las vistas leen la hora actual de la variable de función sgcPlannerNow, que una prueba puede sustituir para congelar el reloj.

Plataformas

Destino	Compatibilidad
Delphi VCL	Delphi 7 a Delphi 13, Win32 y Win64.
Delphi FireMonkey	Delphi XE6 a Delphi 13 (SGC_PLANNER_FMX), toda plataforma que admita FireMonkey.
Linux64	Delphi 10.2 y posteriores para las unidades neutrales respecto al framework (SGC_PLANNER_LINUX): los datos, los motores, JSON, iCalendar, XML de MS Project y el renderer sin ventana.

C++Builder	C++Builder 2007 a C++Builder 13, con los mismos componentes. Consulte Inicio rápido C++Builder.

Delphi 7 y Delphi 2007 tienen cadenas Ansi, de modo que los componentes iCalendar y XML de MS Project convierten el texto UTF-8 de los archivos a la página de códigos del sistema, y los caracteres fuera de esa página de códigos se pierden.

Rendimiento

Son mediciones de builds release de Win64 durante el desarrollo, a 1920x1080 y escala 1. Win32 es entre 1,5 y 1,7 veces más lento.

Escenario	Win64
Programar automáticamente 10.000 tareas (1.000 resúmenes, 3 niveles) con 15.000 vínculos y 30 festivos, ruta crítica incluida	11 ms
Fotograma del Gantt al desplazar 10.000 tareas con 15.000 vínculos	11 ms
Diseño del Kanban de 5.000 tarjetas en 8 columnas y 6 carriles	5 ms
Vista de semana del Scheduler con 2.000 elementos con hora, diseño	1 ms
Cargar 10.000 filas con TsgcPlannerSource_DataSet	26 ms
Guardar y cargar JSON de 10.000 elementos con 15.000 vínculos	48 ms y 56 ms
Importar 5.000 VEVENT con zonas horarias desde un archivo .ics	84 ms
Importar 10.000 tareas con 15.000 vínculos desde XML de MS Project	264 ms

Consulte también

- Ediciones
- TsgcPlannerData
- TsgcPlannerRenderer
- Plataformas
- Datos compartidos
- Vistas sincronizadas

Ediciones

sgcPlanner se compila desde el mismo código fuente para todas las ediciones.

Edición	Incluye
Registered	Todo: las tres vistas para VCL y FMX, el objeto de datos, el calendario laboral, el arrastrar y soltar, los resúmenes, los hitos, el progreso, el origen DataSet, JSON, la salida PNG y SVG sin ventana y las características avanzadas que se enumeran más abajo. La página de la paleta SGC Planner muestra 11 componentes.
Trial	Las mismas características que Registered. Un build compilado con SGC_TRIAL muestra un mensaje de que es una versión de prueba cuando se crea un componente después de que haya pasado la ventana de lanzamiento indicada en el build. Nada más cambia.
Community	Todo salvo las características avanzadas: las vistas del Scheduler de día, semana laboral, semana, mes y agenda con agrupación por recurso, el Gantt con su árbol de tareas, resúmenes, hitos y progreso, el tablero Kanban con columnas de estado, el arrastrar y soltar en todas las vistas, el origen DataSet, JSON y el renderer. La paleta muestra 9 componentes, sin TsgcPlannerICalendar ni TsgcPlannerMSProject, y no hay aviso de prueba.

Las características avanzadas

Las características avanzadas se compilan bajo el define SGC_PLANNER_ADVANCED, que sgcVer.inc desactiva en la edición Community. Seis unidades no forman parte de la distribución Community: sgcPlanner_Recurrence, sgcPlanner_AutoSchedule, sgcPlanner_Scheduler_Timeline, sgcPlanner_XML, sgcPlanner_ICal y sgcPlanner_MSProject. Las propiedades de las vistas se publican en todas las ediciones, de modo que en Community se conservan pero no tienen efecto:

Característica	En la edición Community
Recurrencia	GetOccurrences devuelve un elemento recurrente una vez, en su propio Start. RecurrenceRule y RecurrenceExceptions se siguen almacenando y guardando. OnRecurrenceEdit y OnRecurrenceDelete del Scheduler nunca se disparan.
Dependencias y programación automática	Los vínculos se almacenan y se guardan, pero el Gantt no dibuja flechas y no tiene arrastrar para vincular. ShowDependencies y AutoSchedule no tienen efecto, y Schedule y SetBaseline del Gantt provocan EsgcPlannerError.
Ruta crítica y líneas base	ShowCriticalPath y ShowBaseline no tienen efecto. BaselineStart y BaselineFinish solo se almacenan.
Timeline de recursos	La vista psvTimeline del Scheduler dibuja un aviso centrado.
Carriles y límites de WIP	SwimlaneBy actúa como pslNone. WIPLimit no muestra contador y nunca bloquea una tarjeta.
iCalendar y XML de MS Project	TsgcPlannerICalendar y TsgcPlannerMSProject no existen.

La edición es una elección en tiempo de compilación en sgcVer.inc. SGC_COMMUNITY también desactiva SGC_TRIAL. El código que usa TsgcPlannerICalendar, TsgcPlannerMSProject u otra clase de las seis unidades, y que también debe compilar con Community, lo pone dentro de {$IFDEF SGC_PLANNER_ADVANCED}. Un formulario que contiene uno de estos dos componentes no carga en un build Community.

Consulte también

- Arquitectura
- Introducción
- Características por edición
- Instalación

Instalación

sgcPlanner forma parte de la distribución sgcUI. Un instalador cubre todos los productos sgcUI e instala solo los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcPlanner** y los demás productos que desee. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos, compila los paquetes en su máquina para cada IDE seleccionado en orden de dependencia e instala el paquete de tiempo de diseño. Los componentes aparecen en la página de la paleta **SGC Planner**.

Qué incluye sgcPlanner

sgcUICore, la base compartida de los productos visuales, se instala automáticamente con sgcPlanner. No tiene casilla propia.

Carpetas instaladas

Cada producto se instala en una sola carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la Trial y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de cada producto instalado, sgcPlanner y sgcUICore incluidos. Es la única ruta de biblioteca de sgcUI.
`Packages`	Los paquetes de cada producto instalado, un conjunto por producto y versión de IDE.
`Lib<V>`	Las unidades compiladas para cada versión de IDE y plataforma, por ejemplo `LibD13` .
`Demos\Planner`	Las demos de sgcPlanner. Consulte Demos.

El instalador añade `sgcUI\Source` a la ruta de biblioteca y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas.

Trial y Community

- **Trial**: el instalador Trial de sgcUI instala sgcPlanner en modo de prueba, con todas las características.
- **Community**: el instalador Community gratuito de sgcUI incluye sgcPlanner sin las características avanzadas (recurrencia, programación automática, flechas de dependencia, ruta crítica, líneas base, Timeline de recursos, carriles, límites de WIP, iCalendar y XML de MS Project). Consulte Ediciones.

Compilar los paquetes manualmente

Para compilar los paquetes usted mismo, ábralos para su versión de IDE desde `sgcUI\Packages`. `<V>` es el sufijo de la versión de Delphi, por ejemplo `sgcPlannerD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (runtime).
`sgcPlanner<V>`	El paquete runtime de sgcPlanner: los datos, los motores, las vistas, los controles VCL y, desde Delphi XE6, los controles FireMonkey. Requiere rtl, vcl, dbrtl, fmx (XE6 y posteriores) y sgcUICore.
`dclsgcPlanner<V>`	El paquete de tiempo de diseño que registra los componentes en la página de la paleta SGC Planner.

Compílelos en orden de dependencia: `sgcUICore`, luego `sgcPlanner`, luego `dclsgcPlanner`. Después instale `dclsgcPlanner` y añada `sgcUI\Source` a la ruta de biblioteca de su versión de Delphi (Herramientas > Opciones > Lenguaje > Opciones de Delphi > Biblioteca > Ruta de biblioteca). Para C++Builder, abra el proyecto `.cbproj` de su versión, por ejemplo `sgcPlannerC13.cbproj`, y compílelo. Los encabezados `.hpp` generados aparecen junto al paquete compilado.

Consulte también

- Plataformas
- Ediciones
- Inicio rápido VCL

Plataformas

sgcPlanner es un único árbol de código fuente para todos los IDE y plataformas. Los datos, los motores y los modelos de vista usan solo la RTL y sgcUICore, de modo que son el mismo código en todos los destinos; solo los seis controles están ligados a un framework.

Destino	Compatibilidad
Delphi VCL	Delphi 7 a Delphi 13, Win32 y Win64: TsgcVCLScheduler, TsgcVCLGantt y TsgcVCLKanban.
Delphi FireMonkey	Delphi XE6 a Delphi 13 para Windows, macOS, iOS y Android: TsgcFMXScheduler, TsgcFMXGantt y TsgcFMXKanban. Compilados bajo SGC_PLANNER_FMX.
Linux64	Delphi 10.2 y posteriores, compilados bajo SGC_PLANNER_LINUX: los datos, el calendario laboral, los motores, JSON, iCalendar, XML de MS Project y TsgcPlannerRenderer, de modo que un servidor puede producir archivos PNG y SVG de un plan.
C++Builder	C++Builder 2007 a C++Builder 13 con los mismos componentes y los encabezados .hpp generados. Consulte Inicio rápido C++Builder.
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, el Scheduler, el Gantt y el tablero Kanban en cada host y la salida PNG y SVG sin ventana, en .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Unidades por framework

Unidades	Framework
sgcPlanner_Const, sgcPlanner_DateUtils, sgcPlanner_Calendar, sgcPlanner_Data, sgcPlanner_TimeScale, sgcPlanner_Recurrence, sgcPlanner_AutoSchedule, sgcPlanner_JSON, sgcPlanner_XML, sgcPlanner_ICal, sgcPlanner_MSProject	Ninguno. El modelo, los motores y los formatos de archivo.
sgcPlanner_View, sgcPlanner_Scheduler, sgcPlanner_Scheduler_Render, sgcPlanner_Scheduler_Timeline, sgcPlanner_Gantt, sgcPlanner_Gantt_Render, sgcPlanner_Kanban, sgcPlanner_Kanban_Render, sgcPlanner_Renderer	Ninguno. Los modelos de vista y sus dibujantes, que dibujan en un TsgcUICanvas de sgcUICore.
sgcPlanner_Source_DataSet	Solo la unidad DB (dbrtl), ninguna unidad de VCL o FireMonkey.
sgcPlanner_VCL, sgcPlanner_Scheduler_VCL, sgcPlanner_Gantt_VCL, sgcPlanner_Kanban_VCL	VCL, solo Windows.
sgcPlanner_FMX, sgcPlanner_Scheduler_FMX, sgcPlanner_Gantt_FMX, sgcPlanner_Kanban_FMX	FireMonkey, Delphi XE6 y posteriores.

Paquetes

sgcPlanner incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcPlannerD13.dpk para Delphi 13 y sgcPlannerC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcPlanner	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey: los datos, el calendario laboral, los motores, los modelos de vista y sus painters, JSON, iCalendar, XML de MS Project, TsgcPlannerRenderer y TsgcPlannerSource_DataSet.	rtl, dbrtl, sgcUICore
sgcPlannerVCL	Runtime, VCL	sgcPlanner_VCL, sgcPlanner_Scheduler_VCL, sgcPlanner_Gantt_VCL y sgcPlanner_Kanban_VCL (TsgcVCLScheduler, TsgcVCLGantt, TsgcVCLKanban)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcPlanner
sgcPlannerFMX	Runtime, FireMonkey, desde XE6	sgcPlanner_FMX, sgcPlanner_Scheduler_FMX, sgcPlanner_Gantt_FMX y sgcPlanner_Kanban_FMX (TsgcFMXScheduler, TsgcFMXGantt, TsgcFMXKanban)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcPlanner
dclsgcPlanner	Tiempo de diseño, VCL	Registra TsgcPlannerData, TsgcVCLScheduler, TsgcVCLGantt, TsgcVCLKanban, TsgcPlannerRenderer, TsgcPlannerSource_DataSet y, fuera de la edición Community, TsgcPlannerICalendar y TsgcPlannerMSProject en la página de paleta **SGC Planner**.	designide, rtl, vcl, sgcPlanner, sgcPlannerVCL
dclsgcPlannerFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXScheduler, TsgcFMXGantt y TsgcFMXKanban en la página de paleta **SGC Planner**.	designide, rtl, fmx, sgcPlanner, sgcPlannerFMX, dclsgcPlanner
sgcPlannerWS, dclsgcPlannerWS	Runtime y tiempo de diseño, puente de sgcWebSockets	TsgcPlannerGoogleSync y, fuera de la edición Community, TsgcPlannerICalSubscription, independientes del framework, en la página de paleta **SGC Planner**.	rtl, sgcPlanner, sgcWebSockets

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcPlannerFMX requiere dclsgcPlanner, así que instale primero dclsgcPlanner.

Versiones del IDE y plataformas de destino

Delphi	sgcPlanner	sgcPlannerVCL	sgcPlannerFMX	dclsgcPlanner, dclsgcPlannerFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32

11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE. sgcPlannerWS se compila para las mismas plataformas que sgcPlanner, y dclsgcPlannerWS para las mismas plataformas que dclsgcPlanner.

C++Builder	sgcPlanner, sgcPlannerVCL	sgcPlannerFMX	dclsgcPlanner, dclsgcPlannerFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcPlanner)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcPlanner)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcPlannerC13.cbproj, sgcPlannerVCLC13.cbproj, sgcPlannerFMXC13.cbproj, dclsgcPlannerC13.cbproj y dclsgcPlannerFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. sgcPlannerWS y dclsgcPlannerWS se compilan para las mismas versiones de C++Builder y plataformas que sgcPlanner y dclsgcPlanner, excepto Win64x, porque necesitan una compilación Win64x del paquete sgcWebSockets. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXScheduler, TsgcFMXGantt y TsgcFMXKanban) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcPlannerVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcPlanner_Scheduler_FMX, sgcPlanner_Gantt_FMX o sgcPlanner_Kanban_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcPlanner se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa los datos, el calendario laboral, los motores, JSON, iCalendar, XML de MS Project y TsgcPlannerRenderer, por ejemplo para producir archivos PNG y SVG de un plan. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Notas por plataforma

- **Alta densidad de píxeles.** Los tamaños como RowHeight, TreeWidth y SlotHeight son en DIP. Los controles VCL los escalan con el PPI del monitor, los controles FireMonkey con la escala del formulario y el renderer con su propiedad Scale.
- **Táctil.** Los controles tratan la entrada táctil como entrada de ratón, de modo que un dedo arrastra una tarjeta, una barra o un evento. Los controles VCL conservan sus propios gestos de desplazamiento, de modo que un cuadro de desplazamiento padre no se apodera de un arrastre.
- **Configuración regional.** Los nombres de días y meses, los formatos de fecha y hora y el separador decimal proceden de sgcPlannerLocale, que se inicializa desde la máquina. Un servidor Linux o una máquina con otro idioma muestra por tanto otras etiquetas. Defina las propiedades de sgcPlannerLocale al inicio para obtener la misma salida en todas partes.
- **Delphi 7 y Delphi 2007.** Estas versiones tienen cadenas Ansi, de modo que los lectores de JSON, iCalendar y XML de MS Project convierten el texto UTF-8 a la página de códigos del sistema, y los caracteres fuera de ella se pierden.
- **Ediciones.** SGC_PLANNER_ADVANCED, que sgcVer.inc desactiva en la edición Community, decide las características avanzadas en todas las plataformas. Consulte Características por edición.

Consulte también

- Arquitectura
- Instalación
- Renderizado sin ventana

Inicio rápido VCL

Instale los paquetes de sgcPlanner para su versión de Delphi. Los componentes están en la página de la paleta **SGC Planner**.

En tiempo de diseño

1. Suelte un **TsgcPlannerData** en un formulario. Añada estados a su colección Statuses (por ejemplo todo, doing y done), personas a Resources y tareas a Items: un Id, un Title, un Start, un Finish y un Status para cada una.
2. Suelte un **TsgcVCLGantt**, defina Align como alTop y defina su propiedad Data con el objeto de datos.
3. Suelte un **TsgcVCLKanban**, defina Align como alRight y defina Data. El tablero muestra una columna por estado.
4. Suelte un **TsgcVCLScheduler**, defina Align como alClient, defina Data y elija una View, por ejemplo psvWorkWeek.
5. Ejecute la aplicación. Arrastre una tarjeta a otra columna, una barra en el Gantt o un evento en el Scheduler: las tres vistas muestran el cambio al instante, porque comparten los datos.

En el código

Rellene los datos entre BeginUpdate y EndUpdate, para que cada vista reciba una notificación. El Finish de un elemento es exclusivo.

```
uses
  SysUtils, Controls, Forms,
  sgcPlanner_Data, sgcPlanner_Scheduler, sgcPlanner_Gantt_VCL,
  sgcPlanner_Kanban_VCL, sgcPlanner_Scheduler_VCL;

function AddTask(aData: TsgcPlannerData; const aId, aTitle: string;
  aStart, aFinish: TDateTime; const aStatus, aOwner: string): TsgcPlannerItem;
begin
  Result := aData.AddItem(aId);
  Result.Title := aTitle;
  Result.Start := aStart;
  Result.Finish := aFinish; // the finish is exclusive
  Result.Status := aStatus;
  Result.Resources := aOwner;
end;

procedure LoadProject(aData: TsgcPlannerData);
var
  vMonday: TDateTime;
begin
  vMonday := EncodeDate(2026, 10, 5);
  // one notification for the whole load
  aData.BeginUpdate;
  try
    aData.Statuses.AddStatus('todo', 'To do');
    aData.Statuses.AddStatus('doing', 'In progress');
    aData.Statuses.AddStatus('done', 'Done');
    aData.Resources.AddResource('ana', 'Ana Lopez');
    aData.Resources.AddResource('ben', 'Ben Carter');
    AddTask(aData, 'design', 'Design', vMonday + EncodeTime(8, 0, 0, 0),
      vMonday + 2 + EncodeTime(17, 0, 0, 0), 'doing', 'ana').Progress := 50;
    AddTask(aData, 'build', 'Build', vMonday + 3 + EncodeTime(8, 0, 0, 0),
      vMonday + 7 + EncodeTime(17, 0, 0, 0), 'todo', 'ben');
    AddTask(aData, 'review', 'Design review', vMonday + 2 + EncodeTime(14, 0,
      0, 0), vMonday + 2 + EncodeTime(15, 0, 0, 0), 'todo', 'ana,ben');
    // Build starts when Design finishes
    aData.Dependencies.AddDependency('design', 'build');
```

```
  finally
    aData.EndUpdate;
  end;
end;

procedure CreatePlanner(aForm: TForm);
var
  oData: TsgcPlannerData;
  oGantt: TsgcVCLGantt;
  oKanban: TsgcVCLKanban;
  oScheduler: TsgcVCLScheduler;
begin
  oData := TsgcPlannerData.Create(aForm);
  LoadProject(oData);

  // the three views share one data object
  oGantt := TsgcVCLGantt.Create(aForm);
  oGantt.Parent := aForm;
  oGantt.Align := alTop;
  oGantt.Height := 300;
  oGantt.Data := oData;

  oKanban := TsgcVCLKanban.Create(aForm);
  oKanban.Parent := aForm;
  oKanban.Align := alRight;
  oKanban.Width := 560;
  oKanban.Data := oData;

  oScheduler := TsgcVCLScheduler.Create(aForm);
  oScheduler.Parent := aForm;
  oScheduler.Align := alClient;
  oScheduler.View := psvWorkWeek;
  oScheduler.Date := EncodeDate(2026, 10, 5);
  oScheduler.Data := oData;
end;
```

Consulte también

- TsgcPlannerData
- TsgcVCLGantt
- TsgcVCLKanban
- TsgcVCLScheduler
- Demos
- Instalación

Inicio rápido FMX

Los componentes de FireMonkey necesitan Delphi XE6 o posterior. Están en la página de la paleta **SGC Planner** de un formulario FireMonkey, junto con los componentes no visuales.

En tiempo de diseño

1. Suelte un **TsgcPlannerData** y rellénelo como en el Inicio rápido VCL.
2. Suelte un **TsgcFMXGantt**, un **TsgcFMXKanban** y un **TsgcFMXScheduler**, defina su Align y defina la propiedad Data de cada uno con el objeto de datos.

Los controles FMX dibujan sus propias barras de desplazamiento, se desplazan con la rueda del ratón y tratan la entrada táctil como entrada de ratón, de modo que un dedo arrastra una tarjeta o una barra. Sus propiedades, métodos y eventos son los mismos que en VCL.

En el código

```
uses
  System.SysUtils, FMX.Types, FMX.Controls, FMX.Forms,
  sgcPlanner_Data, sgcPlanner_Scheduler, sgcPlanner_Gantt_FMX,
  sgcPlanner_Kanban_FMX, sgcPlanner_Scheduler_FMX;

procedure CreatePlanner(aForm: TForm; aData: TsgcPlannerData);
var
  oGantt: TsgcFMXGantt;
  oKanban: TsgcFMXKanban;
  oScheduler: TsgcFMXScheduler;
begin
  // the three views share one data object, filled as in the VCL quick start
  oGantt := TsgcFMXGantt.Create(aForm);
  oGantt.Parent := aForm;
  oGantt.Align := TAlignLayout.Top;
  oGantt.Height := 300;
  oGantt.Data := aData;

  oKanban := TsgcFMXKanban.Create(aForm);
  oKanban.Parent := aForm;
  oKanban.Align := TAlignLayout.Right;
  oKanban.Width := 560;
  oKanban.Data := aData;

  oScheduler := TsgcFMXScheduler.Create(aForm);
  oScheduler.Parent := aForm;
  oScheduler.Align := TAlignLayout.Client;
  oScheduler.View := psvWorkWeek;
  oScheduler.Data := aData;
end;
```

Consulte también

- TsgcFMXScheduler
- TsgcFMXGantt
- TsgcFMXKanban
- Demos
- Plataformas

Inicio rápido C++Builder

Instale los paquetes de sgcPlanner para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de include y de biblioteca del proyecto y use los componentes de la página de la paleta **SGC Planner** exactamente como en Delphi. El instalador añade ambas rutas al IDE por usted. Los encabezados .hpp y los archivos .lib generados de sgcPlanner y sgcUICore están en esa única carpeta, de modo que sgcUICore no necesita una ruta propia.

Qué tener en cuenta en C++

- Los constructores no tienen argumentos por defecto. Cree un componente no visual sin propietario con NULL: `new TsgcPlannerRenderer(NULL)`.
- Los ejemplos de esta ayuda pasan explícitamente todos los argumentos de un método, por ejemplo `AddStatus(Id, Caption, 0)` y `AddDependency(FromId, ToId, pdkFS, 0)`, de modo que compilan con cualquier versión de los encabezados generados.
- Los eventos son closures, de modo que sus manejadores son funciones miembro __fastcall de una clase. Un parámetro var como aAccept es una referencia bool, y un parámetro const string es un `const String`.
- Las enumeraciones no tienen ámbito: escriba `psvWorkWeek`, `prvGantt` o `pwmBlock`.

En el código

Consulte también

- TsgcPlannerData
- Inicio rápido VCL
- TsgcPlannerRenderer
- Demos
- Plataformas

Demos

Las demos están en `sgcUI\Demos\Planner\Delphi`, una carpeta por demo con un `readme.html` que da los detalles. La mayoría de las demos construyen sus datos con un pequeño generador determinista en torno a la semana del 5 de octubre de 2026.

Demo	Componentes	Qué muestra
ProjectPlanner	TsgcVCLGantt, TsgcVCLKanban, TsgcVCLScheduler	El relanzamiento de un sitio web en tres vistas vinculadas a un TsgcPlannerData. Arrastre una tarjeta de To do a Done y la columna de estado del Gantt, el color de la barra y el color del evento en el Scheduler la siguen. Colores de estado con BarColorMode pbcStatus y una selección compartida mediante SelectedId.
TeamScheduler	TsgcVCLScheduler	Tres semanas de reuniones para cinco personas en las seis vistas (día, semana laboral, semana, mes, agenda y la Timeline de recursos). Un stand-up diario y una sincronización semanal a partir de una RecurrenceRule, arrastrar para crear con OnItemCreate y agrupación con Options.GroupByResource.
ProjectGantt	TsgcVCLGantt, TsgcPlannerMSProject	Un plan de release de software con tres fases EDT construidas a partir de ParentId, arrastrar para vincular con OnLinkCreating, Schedule y AutoSchedule en el calendario laboral, la ruta crítica, una línea base que muestra cuánto se retrasó una barra movida, y abrir y guardar XML de Microsoft Project. Edición Registered.
TaskBoard	TsgcVCLKanban	Un backlog de 26 tarjetas con anchos de columna y límites de WIP en Columns, los modos de WIP de aviso y de bloqueo, carriles por recurso, padre o etiqueta, navegación por teclado y una columna Done plegable.
DataSetPlanner	TsgcVCLGantt, TsgcVCLKanban, TsgcPlannerSource_DataSet	Tres tablas TClientDataSet (tareas, personas y vínculos) con sus propios nombres de campo, asignadas con ItemFields, ResourceFields y DependencyFields. Un registro confirmado en el TDBGrid actualiza su elemento, una tarjeta o una barra movida en una vista se escribe en la tabla, y las escrituras esperan mientras el grid edita un registro. MidasLib está enlazada, de modo que no hace falta midas.dll.
ICalendarSync	TsgcVCLScheduler, TsgcPlannerICalendar	Exportar a .ics con la RRULE de una serie, elementos de todo el día como valores DATE y horas UTC o flotantes con ExportUTC. Importar desde Google Calendar, Outlook o cualquier archivo RFC 5545 con las zonas horarias convertidas a la hora local, reemplazar o combinar con ImportMode, y los avisos de OnImportWarning. Edición Registered.
GoogleCalendarSync	TsgcVCLScheduler, TsgcPlannerGoogleSync	Una vista semanal en sincronización bidireccional con un Google Calendar mediante TsgcHTTPGoogleCloud_Calendar_Client de sgcWebSockets: el id y el secreto de cliente OAuth2 escritos en tiempo de ejecución y nunca guardados, Direction, una elección para OnConflict, Sincronizar ahora con SyncAsync, sincronización automática con AutoSyncInterval y un registro de OnSyncDone, OnConflict y OnError. Edición Registered, requiere sgcWebSockets.

ICalSubscription	TsgcVCLScheduler, TsgcPlannerICalSubscription	Una URL de iCalendar fusionada junto a dos elementos locales: URL, intervalo de actualización, Actualizar con RefreshAsync, actualización automática con Active, eventos del feed de solo lectura mediante OnItemChanging e IsSubscriptionItem y un registro de OnRefreshed, OnError y OnImportWarning. Edición Registered, requiere sgcWebSockets.
PlannerFMX	TsgcFMXGantt, TsgcFMXScheduler, TsgcFMXKanban, TsgcPlannerRenderer	Las tres vistas de FireMonkey sobre un objeto de datos para Windows, macOS, iOS y Android: una tarjeta, una barra o un evento movido en una vista se mueve en las otras dos. Una barra de herramientas con la vista del Scheduler, zoom y ajuste del Gantt, Selected a Done y un tema oscuro, y cada vista renderizada sin ventana con las opciones de su control.
HeadlessReport	TsgcPlannerRenderer	Una aplicación de consola sin unidad de VCL ni de FireMonkey, para Windows y Linux64. Carga un proyecto con sgcPlannerJSONLoadFromFile, o escribe primero el proyecto de ejemplo como project.json, y guarda las páginas scheduler, gantt y kanban como PNG y SVG, tras ScrollToTime, ExpandAll y ZoomToFit sobre las vistas que posee el renderer. Uso: `HeadlessReport [project.json] [output folder]`.

Parámetros de línea de comandos

Cada demo tiene dos parámetros que usa la verificación del build. `/snapshot <file.png>` guarda una imagen de la demo y termina: las demos de vista de VCL renderizan sus vistas con TsgcPlannerRenderer, DataSetPlanner, ICalendarSync y PlannerFMX guardan la ventana (un BMP en Delphi 7), y HeadlessReport escribe la página Gantt del proyecto de ejemplo. `/test [file.txt]`, o `--test`, ejecuta una autoprueba del comportamiento principal de la demo, escribe una línea PASS o FAIL por comprobación y termina con el número de fallos. Las demos de vista manejan sus controles con mensajes reales de ratón y de teclado.
Las demos de VCL se compilan con Delphi 7 a Delphi 13, Win32 y Win64. PlannerFMX se compila con Delphi XE6 a Delphi 13.

Consulte también

- Inicio rápido VCL
- Inicio rápido FMX
- Instalación

TsgcPlannerData

Unidad: sgcPlanner_Data
TsgcPlannerData son los datos compartidos de sgcPlanner, un componente no visual. Contiene los elementos, los recursos, los estados, las dependencias y el calendario laboral, y notifica a cada vista vinculada a él. Un Scheduler, un Gantt, un Kanban, un origen DataSet, los componentes iCalendar y MS Project y el renderer pueden usar a la vez el mismo objeto de datos.

Propiedades

Propiedad	Tipo	Descripción
Items	TsgcPlannerItems	Las tareas, tarjetas y eventos. Consulte Items más abajo.
Resources	TsgcPlannerResources	Las personas o salas: Id, Name, Color e Initials (calculadas a partir de Name cuando están vacías). AddResource(aId, aName) y Find(aId).
Statuses	TsgcPlannerStatuses	Los estados del flujo de trabajo: Id, Caption y Color. Son las columnas predeterminadas del Kanban. AddStatus(aId, aCaption, aColor = 0) y Find(aId).
Dependencies	TsgcPlannerDependencies	Los vínculos entre elementos: FromId, ToId, Kind (pdkFS por defecto, pdkSS, pdkFF, pdkSF) y Lag en minutos laborales, negativo para un adelanto. AddDependency(aFromId, aToId, aKind = pdkFS, aLag = 0).
Calendar	TsgcPlannerCalendar	El calendario laboral: WorkDays (por defecto de lunes a viernes, vacío significa todos los días), WorkTime (por defecto `08:00-12:00,13:00-17:00` , hasta 4 intervalos, vacío significa todo el día), Holidays (Date, Name) y FirstDayOfWeek (por defecto pwdMonday). Métodos IsWorkingDay, IsWorkingTime, WorkingMinutesPerDay, NextWorkingTime, PrevWorkingTime, AddWorkingMinutes y WorkingMinutesBetween.
ProjectStart	TDateTime	El inicio del proyecto para la programación automática. 0 usa el Start más temprano de las tareas. ProjectStartTime es su equivalente TsgcPlannerTime.
SelectedId	string	Public. El Id del elemento seleccionado, compartido por las vistas que tienen SyncSelection.
Version	Int64	Public, solo lectura. Crece con cada cambio salvo un cambio solo de la selección.
UpdateCount	Integer	Public, solo lectura. La profundidad de BeginUpdate.

Items

Un **TsgcPlannerItem** es una tarea en el Gantt, una tarjeta en el Kanban y un evento en el Scheduler.

Propiedad	Tipo	Descripción
Id	string	La clave única del elemento. Los padres, los vínculos, los recursos del origen DataSet y los archivos importados se refieren a ella.
Title, Notes	string	El texto del elemento.
Start, Finish	TDateTime	Las fechas. Finish es exclusivo. StartTime y FinishTime son los valores exactos TsgcPlannerTime.
AllDay	Boolean	Un evento de todo el día, mostrado en la banda de todo el día del Scheduler.
Status	string	El Id de un estado: la columna Kanban de la tarjeta. La coincidencia distingue mayúsculas de minúsculas.
Progress	Double	Porcentaje completado, de 0 a 100.
Color	TsgcUIColor	$AARRGGBB. 0 (por defecto) toma el color del estado, después el del primer recurso y después un color de la paleta elegido a partir del Id.
ParentId	string	El Id del elemento padre. Un elemento con hijos es un elemento resumen.
Tags, Resources	string	Listas separadas por comas: etiquetas e Ids de recursos. TagCount, TagAt, ResourceCount y ResourceAt las leen.
Rank	Integer	La posición de la tarjeta en su columna Kanban.
Milestone, Manual	Boolean	Un hito se dibuja como un rombo. Una tarea manual conserva sus fechas cuando se programa el proyecto.
Constraint, ConstraintDate	TsgcPlannerConstraint, TDateTime	pcASAP (por defecto) o pcSNET, comienzo no antes de ConstraintDate. Lo usa la programación automática.
BaselineStart, BaselineFinish	TDateTime	El plan guardado, 0 cuando no está definido. Para dibujarlo, la edición Registered.
RecurrenceRule, RecurrenceExceptions	string	Una RRULE de la RFC 5545, como `FREQ=WEEKLY;BYDAY=MO,WE`, y una lista separada por comas de fechas o fechas y horas excluidas. Se expande en la edición Registered.
RecurrenceParentId, RecurrenceDate	string, TDateTime	Se define en una ocurrencia modificada: la serie a la que pertenece y el inicio de la ocurrencia que sustituye.
IsSummary, Level, Duration, TotalSlack, Critical		Public, solo lectura. Un resumen tiene hijos, Level es la profundidad en el árbol, Duration los minutos laborales en el calendario, TotalSlack y Critical el resultado del motor de programación.

Métodos

Método	Devuelve	Descripción
BeginUpdate, EndUpdate		Agrupan cambios. Los listeners y OnChange se disparan una vez tras el EndUpdate más externo.

AddItem(aId = '')	TsgcPlannerItem	Añade un elemento con ese Id.
FindItem(aId), IndexOfItem(aId)	TsgcPlannerItem, Integer	Busca un elemento por Id mediante un índice ordenado. nil o -1 cuando no existe.
DeleteItem(aId)	Boolean	Elimina un elemento, sus ocurrencias modificadas y sus vínculos. Sus hijos pasan a su padre y la selección se borra si estaba seleccionado. False para un Id desconocido.
MoveItem(aId, aStatus, aRank)	Boolean	Mueve una tarjeta a un estado y una posición, y renumera los rangos de ambas columnas. Es lo que hace un drop en el Kanban.
ChildrenOf(aId, var aList)	Integer	Los hijos de un elemento, o los elementos raíz cuando el Id está vacío.
UpdateSummaries		Propaga las fechas de las hojas a sus resúmenes, con el progreso ponderado por la duración laboral de las hojas.
GetOccurrences(aStart, aFinish, var aList)	Integer	Las ocurrencias en la ventana semiabierta [aStart, aFinish) como TsgcPlannerTime, ordenadas por inicio. En la edición Registered un elemento recurrente da una ocurrencia por fecha de su regla, menos sus excepciones y sus ocurrencias modificadas.
GetRange(out aStart, aFinish)	Boolean	El inicio más temprano y el final más tardío de los elementos. False cuando no hay ninguno.
ItemChanging(Sender, aItem, aFields)	Boolean	Dispara OnItemChanging y devuelve aAccept. Una vista personalizada lo llama antes de confirmar una edición.
AddListener, RemoveListener(aListener)		Registran un TsgcPlannerChangeEvent que recibe aviso de cada cambio. Lo usan las vistas. Un listener no debe cambiar los datos.
Clear		Elimina los elementos, los recursos, los estados y las dependencias. El calendario y ProjectStart se conservan.
LoadFromJSON, SaveToJSON, LoadFromFile, SaveToFile		Todos los datos como JSON (UTF-8 sin BOM). Un documento que no se puede leer provoca EsgcPlannerError y deja los datos sin cambios.

Eventos

Evento	Tipo	Descripción
OnChange	TsgcPlannerChangeEvent	Tras el EndUpdate más externo de un cambio, después de los listeners: aChange con Kinds, ItemId (cuando cambió exactamente un elemento) y Version. El manejador puede volver a cambiar los datos.
OnItemChanging	TsgcPlannerItemChangingEvent	Antes de que una vista confirme una edición hecha con el ratón: aItem, aFields (un conjunto de pifStart, pifFinish, pifStatus, pifProgress y los demás campos del elemento) y var aAccept. Ponga aAccept a False para cancelar el drop.

Ejemplo: Un calendario laboral, una reunión recurrente y el seguimiento de cambios

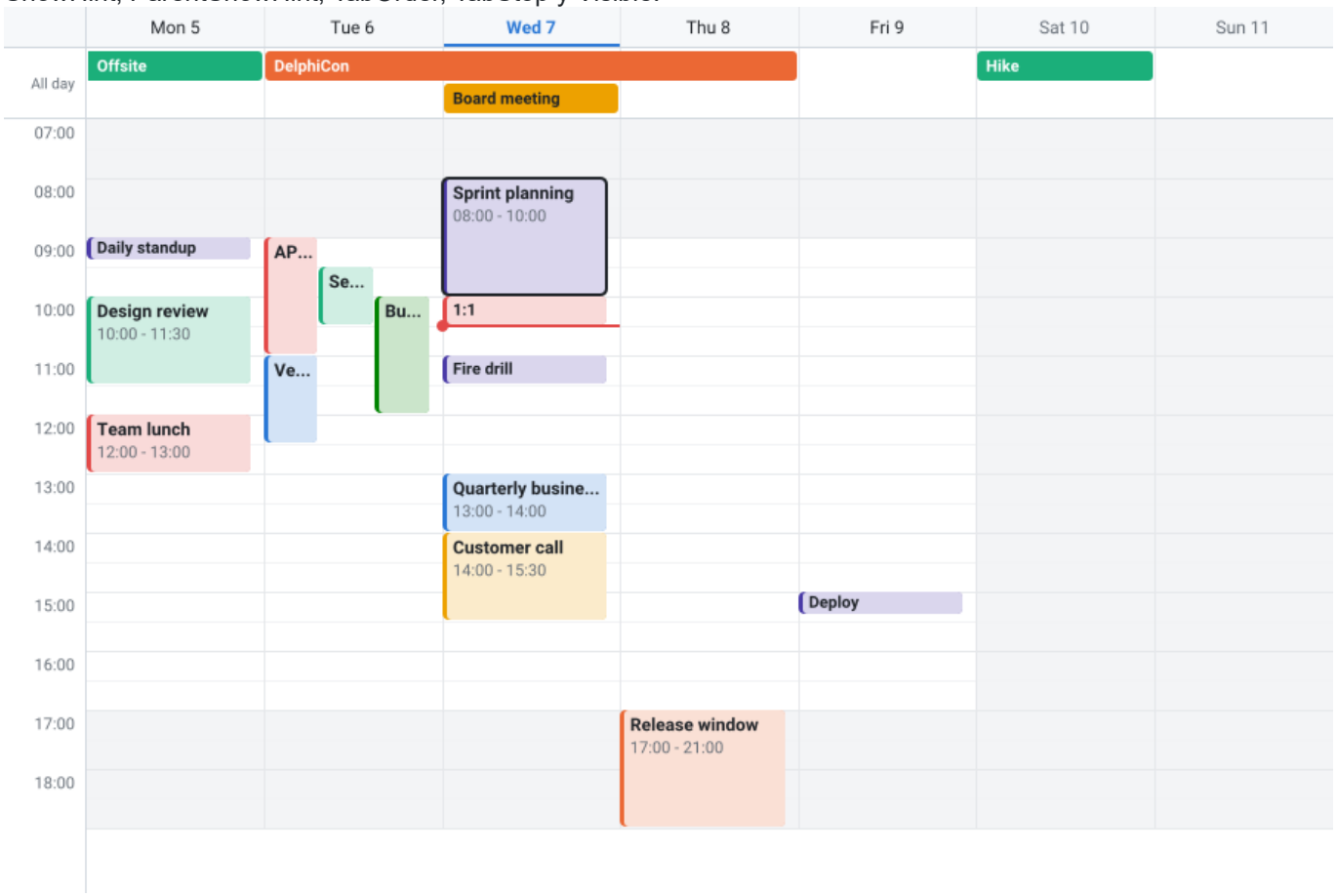
```
uses
  SysUtils, Classes,
  sgcPlanner_DateUtils, sgcPlanner_Data;

type
  TPlannerEvents = class
  public
    Modified: Boolean;
    procedure ItemChanging(Sender: TObject; aItem: TsgcPlannerItem;
      const aFields: TsgcPlannerItemFields; var aAccept: Boolean);
    procedure DataChange(Sender: TObject; const aChange: TsgcPlannerChange);
  end;

procedure TPlannerEvents.ItemChanging(Sender: TObject; aItem: TsgcPlannerItem;
  const aFields: TsgcPlannerItemFields; var aAccept: Boolean);
begin
  // a finished task keeps its dates
  if (aItem.Status = 'done') and ((pifStart in aFields) or
    (pifFinish in aFields)) then
    aAccept := False;
end;

procedure TPlannerEvents.DataChange(Sender: TObject;
  const aChange: TsgcPlannerChange);
begin
  // a selection only change is not an edit
  if aChange.Kinds - [pckSelection] <> [] then
    Modified := True;
end;

function CountItemsOfWeek(aData: TsgcPlannerData; aMonday: TDateTime): Integer;
var
  vList: TsgcPlannerOccurrenceArray;
begin
  // half open window [Monday, next Monday), recurring items expanded
  Result := aData.GetOccurrences(sgcPlannerTimeOf(aMonday),
    sgcPlannerTimeOf(aMonday + 7), vList);
end;

procedure SetupData(aData: TsgcPlannerData; aEvents: TPlannerEvents);
var
  oItem: TsgcPlannerItem;
begin
  aData.Calendar.WorkTime := '08:00-12:00,13:00-17:00';
  aData.Calendar.Holidays.AddHoliday(EncodeDate(2026, 12, 25), 'Christmas');
  aData.OnItemChanging := aEvents.ItemChanging;
  aData.OnChange := aEvents.DataChange;

  // a weekly meeting on Mondays and Wednesdays, ten times
  oItem := aData.AddItem('standup');
  oItem.Title := 'Stand-up';
  oItem.Start := EncodeDate(2026, 10, 5) + EncodeTime(9, 0, 0, 0);
  oItem.Finish := EncodeDate(2026, 10, 5) + EncodeTime(9, 15, 0, 0);
  oItem.RecurrenceRule := 'FREQ=WEEKLY;BYDAY=MO,WE;COUNT=10';

  aData.SaveToFile('project.json');
  aEvents.Modified := False;
end;
```

Consulte también

- Arquitectura
- Inicio rápido VCL
- TsgcPlannerSource_DataSet
- Datos compartidos
- Calendario laboral
- Recurrencia
- Dependencias
- Persistencia JSON

TsgcVCLScheduler

Unidad: sgcPlanner_Scheduler_VCL
TsgcVCLScheduler es el Scheduler de VCL, solo para Windows. Muestra los elementos de un TsgcPlannerData en una vista de día, semana laboral, semana, mes o agenda, con una banda de todo el día, una línea en la hora actual, eventos superpuestos uno al lado del otro y, con Options.GroupByResource, una columna por recurso. En la edición Registered tiene además la Timeline de recursos: una fila por recurso en una escala de tiempo. El usuario crea un evento arrastrando sobre espacio libre, lo mueve y lo redimensiona, y el drop se ajusta a SlotMinutes. Un movimiento a la columna de otro recurso cambia el recurso. Ctrl y la rueda del ratón cambian la altura del slot en las vistas de día. Hereda de TsgcUIVCLControl y publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop y Visible.



Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que muestra el control. Varios controles pueden compartir un objeto de datos. Cuando es nil el control usa un objeto de datos interno propio.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
Colors	TsgcPlannerViewColors	Sustituciones de los colores del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. 0 (por defecto) mantiene el color del tema.
ReadOnly	Boolean	Desactiva todo arrastre y la tecla Supr. Por defecto False.

DragOptions	TsgcPlannerDragOptions	Los arrastres que puede hacer el usuario: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos.
SyncSelection	Boolean	Comparte la selección con las otras vistas de los mismos datos, mediante SelectedId de los datos. Por defecto True.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
BorderStyle	TBorderStyle	bsSingle (por defecto) o bsNone.
View	TsgcPlannerSchedulerViewKind	psvDay, psvWorkWeek (los días laborables del calendario), psvWeek (por defecto), psvMonth (seis semanas), psvAgenda (una lista de AgendaDays días) o psvTimeline (una fila por recurso, edición Registered).
Date	TDateTime	La fecha que muestra la vista. 0 (por defecto) sigue el reloj.
Options	TsgcPlannerSchedulerOptions	DayStartHour y DayEndHour (por defecto 0 y 24), SlotMinutes (30), SlotHeight (20 DIP), ShowAllDayBand (True), ShowCurrentTime (True), GroupByResource (False), MonthMaxItems (0, tantos como quepan), AgendaDays (7) y ResourceRowHeight (40 DIP, Timeline).
TimeScale	TsgcPlannerTimeScale	El eje de tiempo de la Timeline: Zoom (pz5Minutes a pzYears, por defecto pzDays), TickWidth, Origin, Snap, ShowNonWorking, MajorFormat y MinorFormat. El zoom también fija el periodo de la Timeline: un día hasta pzHours, 4 semanas en pzDays, 3 meses en pzWeeks y un año en pzMonths.
SchedulerView	TsgcPlannerSchedulerView	Public, solo lectura. El modelo de vista, con el diseño de los días, las columnas y las entradas.

Métodos

Método	Devuelve	Descripción
GoToday		Establece Date en hoy.
Next, Prior		Mueve Date al periodo siguiente o al anterior de la vista.
ScrollToTime(aTime)		Muestra el día de aTime y desplaza la cuadrícula horaria hasta su hora.
VisibleRange(out aStart, aFinish)		El periodo que muestra la vista.
ItemAt(X, Y)	TsgcPlannerItem	El elemento bajo un punto del control, nil cuando no hay ninguno.
ScrollToItem(aId)	Boolean	Se desplaza hasta que el elemento es visible. False cuando el elemento no se muestra.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		La vista como imagen, la misma imagen que un TsgcPlannerRenderer con los mismos datos, opciones, tamaño y escala. El tamaño es en DIP, 0 es el tamaño del cliente, y una escala de 0 es la escala del control. Se conserva la posición de desplazamiento. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

Eventos

Evento	Tipo	Descripción
OnItemClick, OnItemDblClick	TsgcPlannerI-temEvent	Un clic o un doble clic en un elemento: aItem.
OnItemDelete	TsgcPlannerI-temAcceptE-vent	La tecla Supr sobre el elemento seleccionado: aItem y var aAccept. Ponga aAccept a False para conservar el elemento. Un elemento eliminado se lleva sus vínculos, y sus hijos pasan a su padre.
OnSelection-Change	TNotifyEvent	El elemento seleccionado cambió, en este control o, con SyncSelection, en otra vista de los mismos datos.
OnItemCreate	TsgcPlannerS-chedulerItem-CreateEvent	Un arrastre sobre espacio libre: aStart, aFinish, aResourceId (la columna cuando se agrupa por recurso), var aTitle (por defecto "New event") y var aAccept (por defecto True). El nuevo elemento queda seleccionado.
OnRecurrenceE-dit	TsgcPlanne-rRecurrenceE-ditEvent	Un drop sobre una ocurrencia de una serie recurrente: aItem (la serie), aOccurrence y var aScope. prsOccurrence (por defecto) crea una ocurrencia modificada, prsSeries desplaza toda la serie y prs-Cancel deshace el drop. Edición Registered.
OnRecurrence-Delete	TsgcPlanne-rRecurrenceE-ditEvent	La tecla Supr sobre una ocurrencia de una serie recurrente, antes de borrar nada. prsOccurrence (por defecto) añade la ocurrencia a las excepciones de la serie, prsSeries borra la serie después de OnItem-Delete, prsCancel no borra nada. Edición Registered.
OnDateChange, OnViewChange	TNotifyEvent	Cambió Date o View, también por un clic en "+N more" en la vista de mes, que abre ese día.

Ejemplo: Horario de oficina, columnas de recursos y eventos recurrentes

```
uses
  SysUtils, Classes, Controls, Dialogs,
  sgcPlanner_Data, sgcPlanner_Scheduler, sgcPlanner_Scheduler_VCL;

type
  TSchedulerEvents = class
  public
    procedure ItemCreate(Sender: TObject; const aStart, aFinish: TDateTime;
      const aResourceId: string; var aTitle: string; var aAccept: Boolean);
    procedure RecurrenceEdit(Sender: TObject; aItem: TsgcPlannerItem;
      const aOccurrence: TsgcPlannerOccurrence;
      var aScope: TsgcPlannerRecurrenceScope);
  end;

procedure TSchedulerEvents.ItemCreate(Sender: TObject;
  const aStart, aFinish: TDateTime; const aResourceId: string;
  var aTitle: string; var aAccept: Boolean);
begin
  // no new events on Saturday or Sunday
  aAccept := not (DayOfWeek(aStart) in [1, 7]);
  aTitle := 'Meeting';
end;

procedure TSchedulerEvents.RecurrenceEdit(Sender: TObject;
  aItem: TsgcPlannerItem; const aOccurrence: TsgcPlannerOccurrence;
  var aScope: TsgcPlannerRecurrenceScope);
```

```
begin
  case MessageDlg('Change every occurrence of "' + aItem.Title + '"?',
    mtConfirmation, [mbYes, mbNo, mbCancel], 0) of
    mrYes:
      aScope := prsSeries;
    mrNo:
      aScope := prsOccurrence;
  else
    aScope := prsCancel;
  end;
end;

procedure SetupScheduler(aScheduler: TsgcVCLScheduler;
  aEvents: TSchedulerEvents);
begin
  aScheduler.View := psvWorkWeek;
  aScheduler.Options.DayStartHour := 7;
  aScheduler.Options.DayEndHour := 19;
  aScheduler.Options.SlotMinutes := 15;
  aScheduler.Options.GroupByResource := True;
  aScheduler.OnItemCreate := aEvents.ItemCreate;
  aScheduler.OnRecurrenceEdit := aEvents.RecurrenceEdit;
  // the same question before a delete of a recurring event
  aScheduler.OnRecurrenceDelete := aEvents.RecurrenceEdit;
  aScheduler.GoToday;
  aScheduler.ScrollToTime(Now);
end;
```

Consulte también

- TsgcFMXScheduler
- TsgcPlannerData
- Inicio rápido VCL
- Recurrencia
- Escala de tiempo y zoom
- Arrastrar y soltar
- Vistas sincronizadas

TsgcFMXScheduler

Unidad: sgcPlanner_Scheduler_FMX
TsgcFMXScheduler es el Scheduler de FireMonkey, desde Delphi XE6 (SGC_PLANNER_FMX). Tiene las mismas vistas, propiedades, métodos y eventos que TsgcVCLScheduler. Dibuja sus propias barras de desplazamiento y trata la entrada táctil como entrada de ratón. Hereda de TsgcUIFMXControl y publica las propiedades y eventos de diseño habituales de FireMonkey.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que muestra el control. Varios controles pueden compartir un objeto de datos. Cuando es nil el control usa un objeto de datos interno propio.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
Colors	TsgcPlannerViewColors	Sustituciones de los colores del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. 0 (por defecto) mantiene el color del tema.
ReadOnly	Boolean	Desactiva todo arrastre y la tecla Supr. Por defecto False.
DragOptions	TsgcPlannerDragOptions	Los arrastres que puede hacer el usuario: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos.
SyncSelection	Boolean	Comparte la selección con las otras vistas de los mismos datos, mediante SelectedId de los datos. Por defecto True.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
View	TsgcPlannerSchedulerViewKind	psvDay, psvWorkWeek (los días laborables del calendario), psvWeek (por defecto), psvMonth (seis semanas), psvAgenda (una lista de AgendaDays días) o psvTimeline (una fila por recurso, edición Registered).
Date	TDateTime	La fecha que muestra la vista. 0 (por defecto) sigue el reloj.
Options	TsgcPlannerSchedulerOptions	DayStartHour y DayEndHour (por defecto 0 y 24), SlotMinutes (30), SlotHeight (20 DIP), ShowAllDayBand (True), ShowCurrentTime (True), GroupByResource (False), MonthMaxItems (0, tantos como quepan), AgendaDays (7) y ResourceRowHeight (40 DIP, Timeline).
TimeScale	TsgcPlannerTimeScale	El eje de tiempo de la Timeline: Zoom (pz5Minutes a pzYears, por defecto pzDays), TickWidth, Origin, Snap, ShowNonWorking, MajorFormat y MinorFormat. El zoom también fija el periodo de la Timeline: un día hasta pzHours, 4 semanas en pzDays, 3 meses en pzWeeks y un año en pzMonths.
SchedulerView	TsgcPlannerSchedulerView	Public, solo lectura. El modelo de vista, con el diseño de los días, las columnas y las entradas.

Métodos

Método	Devuelve	Descripción
GoToday		Establece Date en hoy.
Next, Prior		Mueve Date al periodo siguiente o al anterior de la vista.
ScrollToTime(aTime)		Muestra el día de aTime y desplaza la cuadrícula horaria hasta su hora.
VisibleRange(out aStart, aFinish)		El periodo que muestra la vista.
ItemAt(X, Y (Single))	TsgcPlannerItem	El elemento bajo un punto del control, nil cuando no hay ninguno.
ScrollToItem(aId)	Boolean	Se desplaza hasta que el elemento es visible. False cuando el elemento no se muestra.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		La vista como imagen, la misma imagen que un TsgcPlannerRenderer con los mismos datos, opciones, tamaño y escala. El tamaño es en DIP, 0 es el tamaño del cliente, y una escala de 0 es la escala del control. Se conserva la posición de desplazamiento. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

Eventos

Evento	Tipo	Descripción
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Un clic o un doble clic en un elemento: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	La tecla Supr sobre el elemento seleccionado: aItem y var aAccept. Ponga aAccept a False para conservar el elemento. Un elemento eliminado se lleva sus vínculos, y sus hijos pasan a su padre.
OnSelectionChange	TNotifyEvent	El elemento seleccionado cambió, en este control o, con SyncSelection, en otra vista de los mismos datos.
OnItemCreate	TsgcPlannerSchedulerItemCreateEvent	Un arrastre sobre espacio libre: aStart, aFinish, aResourceId (la columna cuando se agrupa por recurso), var aTitle (por defecto "New event") y var aAccept (por defecto True). El nuevo elemento queda seleccionado.
OnRecurrenceEdit	TsgcPlannerRecurrenceEditEvent	Un drop sobre una ocurrencia de una serie recurrente: aItem (la serie), aOccurrence y var aScope. prsOccurrence (por defecto) crea una ocurrencia modificada, prsSeries desplaza toda la serie y prsCancel deshace el drop. Edición Registered.
OnRecurrenceDelete	TsgcPlannerRecurrenceEditEvent	La tecla Supr sobre una ocurrencia de una serie recurrente, antes de borrar nada. prsOccurrence (por defecto) añade la ocurrencia a las excepciones de la serie, prsSeries borra la serie después de OnItemDelete, prsCancel no borra nada. Edición Registered.
OnDateChange, OnViewChange	TNotifyEvent	Cambió Date o View, también por un clic en "+N more" en la vista de mes, que abre ese día.

Ejemplo: La Timeline de recursos guardada como PNG

```
uses
  System.SysUtils, System.Classes, FMX.Types, FMX.Controls,
  sgcPlanner_Data, sgcPlanner_TimeScale, sgcPlanner_Scheduler,
  sgcPlanner_Scheduler_FMX;

procedure ShowTimeline(aScheduler: TsgcFMXScheduler; aData: TsgcPlannerData);
begin
  aScheduler.Align := TAlignLayout.Client;
  aScheduler.Data := aData;
  // one row per resource, four weeks at the day zoom
  aScheduler.View := psvTimeline;
  aScheduler.TimeScale.Zoom := pzDays;
  aScheduler.Options.ResourceRowHeight := 48;
  aScheduler.Date := Date;
  // the same picture as a file, at twice the size
  aScheduler.SaveToPNG('timeline.png', 0, 0, 2);
end;
```

Consulte también

- TsgcVCLScheduler
- Inicio rápido FMX
- Recurrencia
- Escala de tiempo y zoom
- Arrastrar y soltar

TsgcVCLGantt

Unidad: sgcPlanner_Gantt_VCL
TsgcVCLGantt es el diagrama de Gantt de VCL, solo para Windows. Muestra los elementos de un TsgcPlannerData como un árbol de tareas con números EDT y columnas, y un diagrama de barras en una escala de tiempo de dos niveles: barras con su progreso, rombos para los hitos, corchetes para los resúmenes, el tiempo no laboral sombreado y una línea en hoy. El usuario mueve una barra, que conserva su duración laboral, la redimensiona, arrastra su progreso, arrastra desde el extremo de una barra hasta otra barra para crear un vínculo, y abre y cierra los resúmenes. Un movimiento fija un comienzo no antes de la restricción salvo que la tarea sea Manual, y los resúmenes siguen a sus hijos. En la edición Registered los vínculos se dibujan como flechas, y el Gantt puede programar el proyecto, mostrar la ruta crítica y comparar el plan con una línea base. Ctrl y la rueda del ratón hacen zoom en la escala de tiempo. Hereda de TsgcUIVCLControl y publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop y Visible.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que muestra el control. Varios controles pueden compartir un objeto de datos. Cuando es nil el control usa un objeto de datos interno propio.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
Colors	TsgcPlannerViewColors	Sustituciones de los colores del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. 0 (por defecto) mantiene el color del tema.
ReadOnly	Boolean	Desactiva todo arrastre y la tecla Supr. Por defecto False.
DragOptions	TsgcPlannerDragOptions	Los arrastres que puede hacer el usuario: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos.
SyncSelection	Boolean	Comparte la selección con las otras vistas de los mismos datos, mediante SelectedId de los datos. Por defecto True.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
BorderStyle	TBorderStyle	bsSingle (por defecto) o bsNone.
Columns	TsgcPlannerGanttColumns	Las columnas del árbol de tareas. Cada una tiene un Field (pgcId, pgcWBS, pgcTitle, pgcStart, pgcFinish, pgcDuration, pgcProgress, pgcStatus, pgcResources), una Caption (vacía usa el título predeterminado), Width, Visible y Alignment. Las columnas predeterminadas son ID, nombre de la tarea, inicio, fin, duración y progreso. AddColumn(aField, aWidth = 100, aAlignment = taLeftJustify) y SetDefaults.
TreeWidth	Integer	El ancho del árbol de tareas en DIP. Por defecto 550. El usuario arrastra el divisor para cambiarlo.
RowHeight	Integer	El alto de una fila en DIP. Por defecto 28.
TimeScale	TsgcPlannerTimeScale	El eje de tiempo: Zoom (pz5Minutes, pz15Minutes, pzHours, pzDays por defecto, pzWeeks, pzMonths, pzQuarters, pzYears), TickWidth (0 usa el valor por defecto del zoom), Origin, Snap (psNone, psTick por defecto, ps5Min, ps15Min,

		psHour, psDay), ShowNonWorking (True) y MajorFormat y MinorFormat de los dos niveles del encabezado.
DurationUnit	TsgcPlannerDurationUnit	La unidad de la columna de duración: pduMinutes, pduHours, pduDays (por defecto) o pduWeeks, contada en tiempo laboral. El separador decimal es el de la configuración regional del Planner.
BarColorMode	TsgcPlannerBarColorMode	pbcItem (por defecto) colorea una barra como el elemento, pbcStatus según su estado y pbcResource según su primer recurso.
ShowProgress, ShowToday	Boolean	Dibujan el progreso dentro de las barras y una línea en hoy. Por defecto True.
ShowDependencies	Boolean	Dibuja los vínculos como flechas en codo. Por defecto True. Edición Registered.
ShowCriticalPath, ShowBaseline	Boolean	Dibujan las tareas y los vínculos críticos en el color crítico, y la línea base bajo cada barra. Por defecto False. Edición Registered.
AutoSchedule	Boolean	Vuelve a programar el proyecto tras cada edición hecha en el diagrama. Por defecto False. Edición Registered.
GanttView	TsgcPlannerGanttView	Public, solo lectura. El modelo de vista, con las filas, la geometría de las barras y los vínculos.
View	TsgcPlannerView	Public, solo lectura. El modelo de vista que aloja el control.

Métodos

Método	Devuelve	Descripción
Expand(aId), Collapse(aId)	Boolean	Abre o cierra un elemento resumen.
ExpandAll, CollapseAll		Abre o cierra todos los elementos resumen.
ZoomIn, ZoomOut	Boolean	Un nivel de zoom más fino o más grueso, alrededor del centro del diagrama. False en el último nivel.
ZoomToFit	Boolean	Elige el nivel de zoom más fino que muestra todas las filas y se desplaza hasta el primer inicio. False sin filas.
ScrollToTime(aTime)		Desplaza el diagrama para que aTime quede en su borde izquierdo.
Schedule	Boolean	Ejecuta el motor de programación sobre los datos: tareas después de sus predecesoras, desde ProjectStart, en el calendario laboral, con la holgura y la ruta crítica. Un ciclo en los vínculos dispara OnScheduleError, y las tareas del ciclo conservan sus fechas. En Community provoca EsgcPlannerError.
SetBaseline		Copia Start y Finish de cada tarea a BaselineStart y BaselineFinish. En Community provoca EsgcPlannerError.

ItemAt(X, Y)	TsgcPlannerItem	El elemento bajo un punto del control, nil cuando no hay ninguno.
ScrollToItem(aId)	Boolean	Se desplaza hasta que el elemento es visible. False cuando el elemento no se muestra.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		La vista como imagen, la misma imagen que un TsgcPlannerRenderer con los mismos datos, opciones, tamaño y escala. El tamaño es en DIP, 0 es el tamaño del cliente, y una escala de 0 es la escala del control. Se conserva la posición de desplazamiento. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

Eventos

Evento	Tipo	Descripción
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Un clic o un doble clic en un elemento: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	La tecla Supr sobre el elemento seleccionado: aItem y var aAccept. Ponga aAccept a False para conservar el elemento. Un elemento eliminado se lleva sus vínculos, y sus hijos pasan a su padre.
OnSelectionChange	TNotifyEvent	El elemento seleccionado cambió, en este control o, con SyncSelection, en otra vista de los mismos datos.
OnLinkCreating	TsgcPlannerGanttLinkCreatingEvent	Un arrastre desde el extremo de una barra hasta otra barra: aFrom, aTo, var aKind (según los extremos elegidos) y var aAccept. Los vínculos al propio elemento, a un ancestro o a un descendiente y los duplicados se rechazan antes del evento. Edición Registered.
OnScheduleError	TsgcPlannerGanttScheduleErrorEvent	El motor encontró un ciclo: aError y aCycleIds, los Ids de los elementos del ciclo. Sus fechas se conservan.
OnExpandChange	TsgcPlannerGanttExpandEvent	Se abrió o cerró un resumen: aItem y aExpanded. ExpandAll y CollapseAll lo disparan una vez, con aItem nil.

Ejemplo: Columnas EDT, programación automática y la ruta crítica

```
uses
  SysUtils, Classes, Dialogs,
  sgcPlanner_Data, sgcPlanner_TimeScale, sgcPlanner_Gantt,
  sgcPlanner_Gantt_VCL;

type
  TGanttEvents = class
  public
    procedure LinkCreating(Sender: TObject; aFrom, aTo: TsgcPlannerItem;
      var aKind: TsgcPlannerDependencyKind; var aAccept: Boolean);
    procedure ScheduleError(Sender: TObject; const aError: string;
      aCycleIds: TStrings);
  end;

procedure TGanttEvents.LinkCreating(Sender: TObject;
```

```
  aFrom, aTo: TsgcPlannerItem; var aKind: TsgcPlannerDependencyKind;
  var aAccept: Boolean);
begin
  // a milestone only takes finish to start links
  if aTo.Milestone then
    aKind := pdkFS;
  // no links into finished work
  aAccept := aTo.Progress < 100;
end;

procedure TGanttEvents.ScheduleError(Sender: TObject; const aError: string;
  aCycleIds: TStrings);
begin
  ShowMessage(aError + sLineBreak + aCycleIds.CommaText);
end;

procedure SetupGantt(aGantt: TsgcVCLGantt; aEvents: TGanttEvents);
begin
  aGantt.Columns.Clear;
  aGantt.Columns.AddColumn(pgcWBS, 50);
  aGantt.Columns.AddColumn(pgcTitle, 200);
  aGantt.Columns.AddColumn(pgcResources, 120);
  aGantt.Columns.AddColumn(pgcDuration, 70, taRightJustify);
  aGantt.TreeWidth := 440;
  aGantt.BarColorMode := pbcStatus;
  aGantt.TimeScale.Zoom := pzWeeks;
  aGantt.OnLinkCreating := aEvents.LinkCreating;
  aGantt.OnScheduleError := aEvents.ScheduleError;
  // Registered edition: schedule on every edit, show the critical path
  aGantt.AutoSchedule := True;
  aGantt.ShowCriticalPath := True;
  aGantt.ShowBaseline := True;
  aGantt.Schedule;
  aGantt.SetBaseline;
  aGantt.ExpandAll;
  aGantt.ZoomToFit;
end;
```

Consulte también

- TsgcFMXGantt
- TsgcPlannerData
- TsgcPlannerMSProject
- Dependencias
- Programación automática
- Ruta crítica
- Líneas base
- Escala de tiempo y zoom

TsgcFMXGantt

Unidad: sgcPlanner_Gantt_FMX
TsgcFMXGantt es el diagrama de Gantt de FireMonkey, desde Delphi XE6 (SGC_PLANNER_FMX). Tiene el mismo árbol, el mismo diagrama y las mismas propiedades, métodos y eventos que TsgcVCLGantt. Dibuja sus propias barras de desplazamiento y trata la entrada táctil como entrada de ratón. Hereda de TsgcUIFMXControl y publica las propiedades y eventos de diseño habituales de FireMonkey.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que muestra el control. Varios controles pueden compartir un objeto de datos. Cuando es nil el control usa un objeto de datos interno propio.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
Colors	TsgcPlannerViewColors	Sustituciones de los colores del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. 0 (por defecto) mantiene el color del tema.
ReadOnly	Boolean	Desactiva todo arrastre y la tecla Supr. Por defecto False.
DragOptions	TsgcPlannerDragOptions	Los arrastres que puede hacer el usuario: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos.
SyncSelection	Boolean	Comparte la selección con las otras vistas de los mismos datos, mediante SelectedId de los datos. Por defecto True.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
Columns	TsgcPlannerGanttColumns	Las columnas del árbol de tareas. Cada una tiene un Field (pgcId, pgcWBS, pgcTitle, pgcStart, pgcFinish, pgcDuration, pgcProgress, pgcStatus, pgcResources), una Caption (vacía usa el título predeterminado), Width, Visible y Alignment. Las columnas predeterminadas son ID, nombre de la tarea, inicio, fin, duración y progreso. AddColumn(aField, aWidth = 100, aAlignment = taLeftJustify) y SetDefaults.
TreeWidth	Integer	El ancho del árbol de tareas en DIP. Por defecto 550. El usuario arrastra el divisor para cambiarlo.
RowHeight	Integer	El alto de una fila en DIP. Por defecto 28.
TimeScale	TsgcPlannerTimeScale	El eje de tiempo: Zoom (pz5Minutes, pz15Minutes, pzHours, pzDays por defecto, pzWeeks, pzMonths, pzQuarters, pzYears), TickWidth (0 usa el valor por defecto del zoom), Origin, Snap (psNone, psTick por defecto, ps5Min, ps15Min, psHour, psDay), ShowNonWorking (True) y MajorFormat y MinorFormat de los dos niveles del encabezado.
DurationUnit	TsgcPlannerDurationUnit	La unidad de la columna de duración: pduMinutes, pduHours, pduDays (por defecto) o pduWeeks, contada en tiempo laboral. El separador decimal es el de la configuración regional del Planner.
BarColorMode	TsgcPlannerBar-	pbcItem (por defecto) colorea una barra como el elemento, pbcStatus según su estado y pbcResource según su primer recurso.

	ColorMode	
ShowProgress, ShowToday	Boolean	Dibujan el progreso dentro de las barras y una línea en hoy. Por defecto True.
ShowDependencies	Boolean	Dibuja los vínculos como flechas en codo. Por defecto True. Edición Registered.
ShowCriticalPath, ShowBaseline	Boolean	Dibujan las tareas y los vínculos críticos en el color crítico, y la línea base bajo cada barra. Por defecto False. Edición Registered.
AutoSchedule	Boolean	Vuelve a programar el proyecto tras cada edición hecha en el diagrama. Por defecto False. Edición Registered.
GanttView	TsgcPlannerGanttView	Public, solo lectura. El modelo de vista, con las filas, la geometría de las barras y los vínculos.
View	TsgcPlannerView	Public, solo lectura. El modelo de vista que aloja el control.

Métodos

Método	Devuelve	Descripción
Expand(aId), Collapse(aId)	Boolean	Abre o cierra un elemento resumen.
ExpandAll, CollapseAll		Abre o cierra todos los elementos resumen.
ZoomIn, ZoomOut	Boolean	Un nivel de zoom más fino o más grueso, alrededor del centro del diagrama. False en el último nivel.
ZoomToFit	Boolean	Elige el nivel de zoom más fino que muestra todas las filas y se desplaza hasta el primer inicio. False sin filas.
ScrollToTime(aTime)		Desplaza el diagrama para que aTime quede en su borde izquierdo.
Schedule	Boolean	Ejecuta el motor de programación sobre los datos: tareas después de sus predecesoras, desde ProjectStart, en el calendario laboral, con la holgura y la ruta crítica. Un ciclo en los vínculos dispara OnScheduleError, y las tareas del ciclo conservan sus fechas. En Community provoca EsgcPlannerError.
SetBaseline		Copia Start y Finish de cada tarea a BaselineStart y BaselineFinish. En Community provoca EsgcPlannerError.
ItemAt(X, Y (Single))	TsgcPlannerItem	El elemento bajo un punto del control, nil cuando no hay ninguno.
ScrollToItem(aId)	Boolean	Se desplaza hasta que el elemento es visible. False cuando el elemento no se muestra.

SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		La vista como imagen, la misma imagen que un TsgcPlannerRenderer con los mismos datos, opciones, tamaño y escala. El tamaño es en DIP, 0 es el tamaño del cliente, y una escala de 0 es la escala del control. Se conserva la posición de desplazamiento. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

Eventos

Evento	Tipo	Descripción
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Un clic o un doble clic en un elemento: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	La tecla Supr sobre el elemento seleccionado: aItem y var aAccept. Ponga aAccept a False para conservar el elemento. Un elemento eliminado se lleva sus vínculos, y sus hijos pasan a su padre.
OnSelectionChange	TNotifyEvent	El elemento seleccionado cambió, en este control o, con SyncSelection, en otra vista de los mismos datos.
OnLinkCreating	TsgcPlannerGanttLinkCreatingEvent	Un arrastre desde el extremo de una barra hasta otra barra: aFrom, aTo, var aKind (según los extremos elegidos) y var aAccept. Los vínculos al propio elemento, a un ancestro o a un descendiente y los duplicados se rechazan antes del evento. Edición Registered.
OnScheduleError	TsgcPlannerGanttScheduleErrorEvent	El motor encontró un ciclo: aError y aCycleIds, los Ids de los elementos del ciclo. Sus fechas se conservan.
OnExpandChange	TsgcPlannerGanttExpandEvent	Se abrió o cerró un resumen: aItem y aExpanded. ExpandAll y CollapseAll lo disparan una vez, con aItem nil.

Ejemplo: Horas, un ajuste de 15 minutos y una fase abierta

```
uses
  System.SysUtils, FMX.Types, FMX.Controls,
  sgcPlanner_Data, sgcPlanner_TimeScale, sgcPlanner_Gantt,
  sgcPlanner_Gantt_FMX;

procedure SetupGantt(aGantt: TsgcFMXGantt; aData: TsgcPlannerData);
begin
  aGantt.Align := TAlignLayout.Client;
  aGantt.Data := aData;
  aGantt.RowHeight := 32;
  aGantt.DurationUnit := pduHours;
  aGantt.TimeScale.Snap := ps15Min;
  // only the summaries of the first level are open
  aGantt.CollapseAll;
  aGantt.Expand('phase1');
  aGantt.ScrollToTime(Now);
end;
```

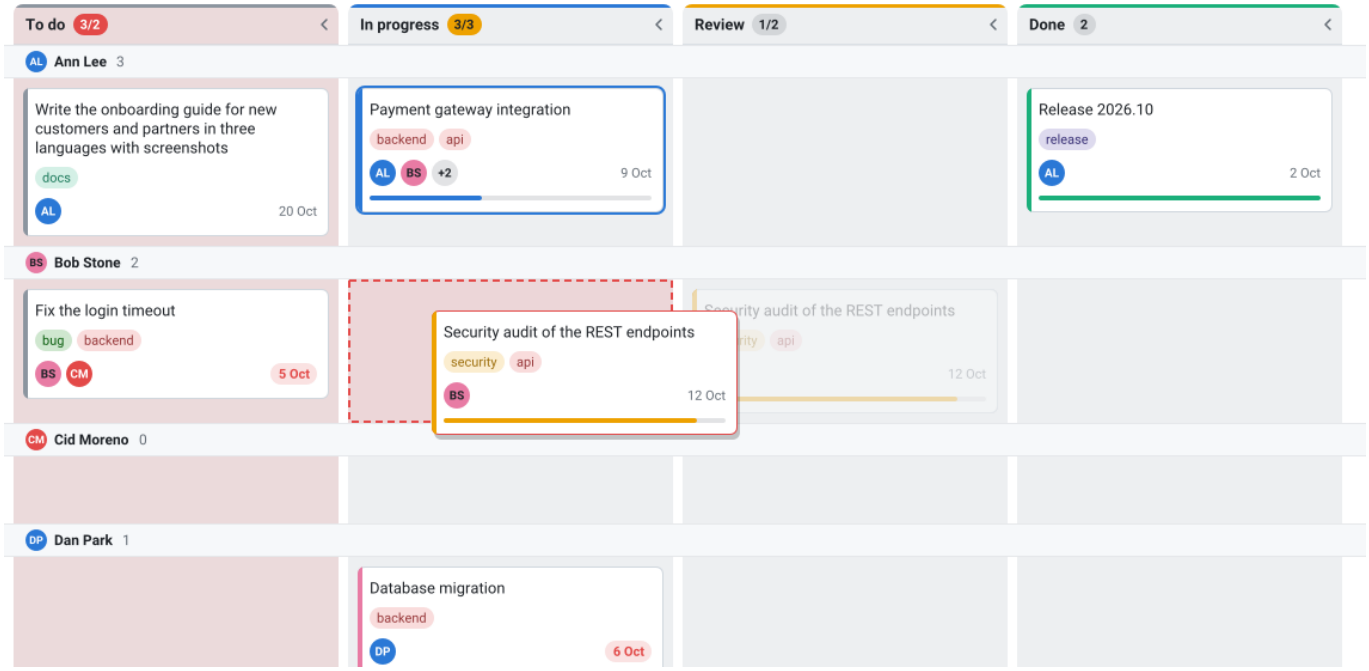
Consulte también

- TsgcVCLGantt
- Inicio rápido FMX
- Dependencias

- Programación automática
- Ruta crítica
- Líneas base
- Escala de tiempo y zoom

TsgcVCLKanban

Unidad: sgcPlanner_Kanban_VCL
TsgcVCLKanban es el tablero Kanban de VCL, solo para Windows. Muestra los elementos de un TsgcPlannerData como tarjetas en un tablero: una columna por estado, con un contador, un límite de WIP opcional y una flecha para plegar, y en la edición Registered carriles por recurso, padre, etiqueta o un evento. Una tarjeta muestra su título en varias líneas, sus etiquetas, las iniciales de sus recursos, su fecha de vencimiento y su progreso. El usuario arrastra una tarjeta a otra posición, columna o carril, y el drop cambia el Status y el Rank del elemento. Los elementos resumen y las ocurrencias modificadas no se muestran como tarjetas. Con pdoReorder y sin pdoMove en DragOptions, un arrastre solo cambia el orden de las tarjetas de una columna. Hereda de TsgcUIVCLControl y publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop y Visible.



Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que muestra el control. Varios controles pueden compartir un objeto de datos. Cuando es nil el control usa un objeto de datos interno propio.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
Colors	TsgcPlannerViewColors	Sustituciones de los colores del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. 0 (por defecto) mantiene el color del tema.
ReadOnly	Boolean	Desactiva todo arrastre y la tecla Supr. Por defecto False.
DragOptions	TsgcPlannerDragOptions	Los arrastres que puede hacer el usuario: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos.
SyncSelection	Boolean	Comparte la selección con las otras vistas de los mismos datos, mediante SelectedId de los datos. Por defecto True.

MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
BorderStyle	TBorderStyle	bsSingle (por defecto) o bsNone.
ColumnSource	TsgcPlannerKanbanColumnSource	pkcStatuses (por defecto) muestra una columna por estado de los datos, pkcList muestra solo la colección Columns.
Columns	TsgcPlannerKanbanColumns	Las columnas: StatusId, Caption y Color (vacíos o 0 toman el estado), WIPLimit (0 ninguno), Collapsed y Width (0 por defecto). Con pkcStatuses una entrada solo añade ajustes a la columna de su estado. AddColumn(aStatusId, aCaption = '', aWIPLimit = 0) y Find(aStatusId).
ShowUnknownColumn	Boolean	Muestra una columna Otros para las tarjetas cuyo estado no tiene columna. Por defecto True. No se puede soltar una tarjeta en ella.
SwimlaneBy	TsgcPlannerSwimlaneBy	pslNone (por defecto), pslResource (un carril por recurso, un drop en otro carril sustituye al primer recurso), pslParent (un carril por padre, un drop establece ParentId), pslTag (un carril por etiqueta) o pslEvent (OnGetSwimlane da el carril). Edición Registered.
WIPMode	TsgcPlannerWIPMode	pwmWarn (por defecto) solo colorea el contador de una columna en o por encima de su WIPLimit, pwmBlock también rechaza una tarjeta que superaría el límite. Reordenar dentro de una columna está siempre permitido. Edición Registered.
CardOptions	TsgcPlannerKanbanCardOptions	ShowTags, ShowAssignees, ShowDueDate y ShowProgress (por defecto True) y MaxTitleLines. La fecha de vencimiento es el Finish del elemento, mostrada en rojo cuando ha pasado y la tarjeta no está completa.
KanbanView	TsgcPlannerKanbanView	Public, solo lectura. El modelo de vista, con las columnas, los carriles y las tarjetas.
View	TsgcPlannerView	Public, solo lectura. El modelo de vista que aloja el control.

Métodos

Método	Devuelve	Descripción
CollapseColumn(aStatusId), ExpandColumn(aStatusId)	Boolean	Pliega una columna en un raíl estrecho o la abre de nuevo. El usuario hace clic en la flecha del encabezado.
SelectNextCard(aKey)	Boolean	Mueve la selección con un código de tecla: las flechas recorren las tarjetas, las columnas y los carriles, Inicio y Fin van a la primera y a la última tarjeta.
ItemAt(X, Y)	TsgcPlannerItem	El elemento bajo un punto del control, nil cuando no hay ninguno.
ScrollToItem(aId)	Boolean	Se desplaza hasta que el elemento es visible. False cuando el elemento no se muestra.

SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		La vista como imagen, la misma imagen que un TsgcPlannerRenderer con los mismos datos, opciones, tamaño y escala. El tamaño es en DIP, 0 es el tamaño del cliente, y una escala de 0 es la escala del control. Se conserva la posición de desplazamiento. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

Eventos

Evento	Tipo	Descripción
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Un clic o un doble clic en un elemento: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	La tecla Supr sobre el elemento seleccionado: aItem y var aAccept. Ponga aAccept a False para conservar el elemento. Un elemento eliminado se lleva sus vínculos, y sus hijos pasan a su padre.
OnSelectionChange	TNotifyEvent	El elemento seleccionado cambió, en este control o, con SyncSelection, en otra vista de los mismos datos.
OnCardMoving	TsgcPlannerKanbanCardMovingEvent	Antes de un drop: aItem, aToStatus, aToLane y var aAccept. Se dispara antes de OnItemChanging de los datos.
OnCardMoved	TsgcPlannerItemEvent	Después de confirmar un drop: aItem con su nuevo estado y rango.
OnGetSwimlane	TsgcPlannerKanbanSwimlaneEvent	Con pslEvent: aItem y var aKey y aCaption, el carril de la tarjeta.
OnColumnCollapse	TsgcPlannerKanbanColumnEvent	Se plegó o se abrió una columna: aStatusId y aCollapsed.

Ejemplo: Un límite de WIP, carriles por persona y una regla para Done

```
uses
  SysUtils, Classes, Dialogs,
  sgcPlanner_Data, sgcPlanner_Kanban, sgcPlanner_Kanban_VCL;

type
  TBoardEvents = class
  public
    procedure CardMoving(Sender: TObject; aItem: TsgcPlannerItem;
      const aToStatus, aToLane: string; var aAccept: Boolean);
    procedure CardMoved(Sender: TObject; aItem: TsgcPlannerItem);
  end;

procedure TBoardEvents.CardMoving(Sender: TObject; aItem: TsgcPlannerItem;
  const aToStatus, aToLane: string; var aAccept: Boolean);
begin
  // only finished work goes to Done
  if (aToStatus = 'done') and (aItem.Progress < 100) then
    aAccept := False;
end;
```

```
procedure TBoardEvents.CardMoved(Sender: TObject; aItem: TsgcPlannerItem);
begin
  ShowMessage(Format('%s is now %s', [aItem.Title, aItem.Status]));
end;

procedure SetupBoard(aKanban: TsgcVCLKanban; aEvents: TBoardEvents);
begin
  // one column per status, In progress takes at most 3 cards
  aKanban.ColumnSource := pkcStatuses;
  aKanban.Columns.AddColumn('doing', '', 3);
  aKanban.WIPMode := pwmBlock;
  // one lane per person (Registered edition)
  aKanban.SwimlaneBy := pslResource;
  aKanban.CardOptions.MaxTitleLines := 2;
  aKanban.CardOptions.ShowDueDate := True;
  aKanban.OnCardMoving := aEvents.CardMoving;
  aKanban.OnCardMoved := aEvents.CardMoved;
end;
```

Consulte también

- TsgcFMXKanban
- TsgcPlannerData
- Inicio rápido VCL
- Carriles y límites de WIP
- Arrastrar y soltar
- Temas

TsgcFMXKanban

Unidad: sgcPlanner_Kanban_FMX
TsgcFMXKanban es el tablero Kanban de FireMonkey, desde Delphi XE6 (SGC_PLANNER_FMX). Tiene el mismo tablero y las mismas propiedades, métodos y eventos que TsgcVCLKanban. Dibuja sus propias barras de desplazamiento y trata la entrada táctil como entrada de ratón, de modo que un dedo arrastra una tarjeta. Hereda de TsgcUIFMXControl y publica las propiedades y eventos de diseño habituales de FireMonkey.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que muestra el control. Varios controles pueden compartir un objeto de datos. Cuando es nil el control usa un objeto de datos interno propio.
Theme	TsgcUITheme	utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
Colors	TsgcPlannerViewColors	Sustituciones de los colores del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. 0 (por defecto) mantiene el color del tema.
ReadOnly	Boolean	Desactiva todo arrastre y la tecla Supr. Por defecto False.
DragOptions	TsgcPlannerDragOptions	Los arrastres que puede hacer el usuario: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos.
SyncSelection	Boolean	Comparte la selección con las otras vistas de los mismos datos, mediante SelectedId de los datos. Por defecto True.
MaxFPS	Integer	El mayor número de fotogramas por segundo que dibuja el control.
ColumnSource	TsgcPlannerKanbanColumnSource	pkcStatuses (por defecto) muestra una columna por estado de los datos, pkcList muestra solo la colección Columns.
Columns	TsgcPlannerKanbanColumns	Las columnas: StatusId, Caption y Color (vacíos o 0 toman el estado), WIPLimit (0 ninguno), Collapsed y Width (0 por defecto). Con pkcStatuses una entrada solo añade ajustes a la columna de su estado. AddColumn(aStatusId, aCaption = '', aWIPLimit = 0) y Find(aStatusId).
ShowUnknownColumn	Boolean	Muestra una columna Otros para las tarjetas cuyo estado no tiene columna. Por defecto True. No se puede soltar una tarjeta en ella.
SwimlaneBy	TsgcPlannerSwimlaneBy	pslNone (por defecto), pslResource (un carril por recurso, un drop en otro carril sustituye al primer recurso), pslParent (un carril por padre, un drop establece ParentId), pslTag (un carril por etiqueta) o pslEvent (OnGetSwimlane da el carril). Edición Registered.
WIPMode	TsgcPlannerWIPMode	pwmWarn (por defecto) solo colorea el contador de una columna en o por encima de su WIPLimit, pwmBlock también rechaza una tarjeta que superaría el límite. Reordenar dentro de una columna está siempre permitido. Edición Registered.

CardOptions	TsgcPlannerKanbanCardOptions	ShowTags, ShowAssignees, ShowDueDate y ShowProgress (por defecto True) y MaxTitleLines. La fecha de vencimiento es el Finish del elemento, mostrada en rojo cuando ha pasado y la tarjeta no está completa.
KanbanView	TsgcPlannerKanbanView	Public, solo lectura. El modelo de vista, con las columnas, los carriles y las tarjetas.
View	TsgcPlannerView	Public, solo lectura. El modelo de vista que aloja el control.

Métodos

Método	Devuelve	Descripción
CollapseColumn(aStatusId), ExpandColumn(aStatusId)	Boolean	Pliega una columna en un raíl estrecho o la abre de nuevo. El usuario hace clic en la flecha del encabezado.
SelectNextCard(aKey)	Boolean	Mueve la selección con un código de tecla: las flechas recorren las tarjetas, las columnas y los carriles, Inicio y Fin van a la primera y a la última tarjeta.
ItemAt(X, Y (Single))	TsgcPlannerItem	El elemento bajo un punto del control, nil cuando no hay ninguno.
ScrollToItem(aId)	Boolean	Se desplaza hasta que el elemento es visible. False cuando el elemento no se muestra.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		La vista como imagen, la misma imagen que un TsgcPlannerRenderer con los mismos datos, opciones, tamaño y escala. El tamaño es en DIP, 0 es el tamaño del cliente, y una escala de 0 es la escala del control. Se conserva la posición de desplazamiento. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

Eventos

Evento	Tipo	Descripción
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Un clic o un doble clic en un elemento: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	La tecla Supr sobre el elemento seleccionado: aItem y var aAccept. Ponga aAccept a False para conservar el elemento. Un elemento eliminado se lleva sus vínculos, y sus hijos pasan a su padre.
OnSelectionChange	TNotifyEvent	El elemento seleccionado cambió, en este control o, con SyncSelection, en otra vista de los mismos datos.
OnCardMoving	TsgcPlannerKanbanCardMovingEvent	Antes de un drop: aItem, aToStatus, aToLane y var aAccept. Se dispara antes de OnItemChanging de los datos.

OnCardMoved	TsgcPlannerItemEvent	Después de confirmar un drop: aItem con su nuevo estado y rango.
OnGetSwimlane	TsgcPlannerKanbanSwimlaneEvent	Con pslEvent: aItem y var aKey y aCaption, el carril de la tarjeta.
OnColumnCollapse	TsgcPlannerKanbanColumnEvent	Se plegó o se abrió una columna: aStatusId y aCollapsed.

Ejemplo: Una lista fija de columnas y una columna Done plegada

```
uses
  System.SysUtils, System.Classes, FMX.Types, FMX.Controls,
  sgcPlanner_Data, sgcPlanner_Kanban, sgcPlanner_Kanban_FMX;

procedure SetupBoard(aKanban: TsgcFMXKanban; aData: TsgcPlannerData);
begin
  aKanban.Align := TAlignLayout.Client;
  aKanban.Data := aData;
  // a fixed list of columns instead of the statuses
  aKanban.ColumnSource := pkcList;
  aKanban.Columns.Clear;
  aKanban.Columns.AddColumn('todo', 'Backlog');
  aKanban.Columns.AddColumn('doing', 'This week', 5);
  aKanban.Columns.AddColumn('done', 'Shipped');
  aKanban.ShowUnknownColumn := False;
  aKanban.CardOptions.ShowTags := True;
  // start with Shipped folded into a rail
  aKanban.CollapseColumn('done');
end;
```

Consulte también

- TsgcVCLKanban
- Inicio rápido FMX
- Carriles y límites de WIP
- Arrastrar y soltar

TsgcPlannerSource_DataSet

Unidad: sgcPlanner_Source_DataSet
TsgcPlannerSource_DataSet vincula un TsgcPlannerData a TDataSets: DataSet contiene los elementos, ResourceDataSet los recursos y DependencyDataSet las dependencias. Solo necesita la unidad DB, de modo que funciona con cualquier dataset, en VCL, FireMonkey o en un servicio. Forma parte de todas las ediciones.

Carga

- Active True carga cada DataSet abierto, y abrir un DataSet mientras Active lo carga. Una carga sustituye los elementos, los recursos y las dependencias en una sola actualización, de modo que cada vista vinculada recibe una notificación. El registro actual de cada DataSet se restaura después.
- Las fechas proceden de campos de fecha y hora, de campos numéricos o de texto ISO 8601. Los booleanos proceden de campos booleanos o numéricos o de texto (1, T, True, Y, Yes). Los colores proceden de campos numéricos (ARGB) o de texto (#RRGGBB, #AARRGGBB, $AARRGGBB). Un nulo da el valor por defecto, y un Finish nulo da Finish = Start, o la siguiente medianoche para un elemento de todo el día.
- Sin un DependencyDataSet, DependsOnField contiene los Ids separados por comas de las predecesoras del elemento, leídos como vínculos de fin a comienzo.

Actualizaciones en vivo y escritura de vuelta

- Con LiveUpdates, una edición o inserción confirmada actualiza o añade solo ese elemento, y un cambio en el número de registros (un borrado, un filtro, un refresh) sincroniza los elementos por Id. El desplazamiento y las ediciones sin confirmar no cambian nada.
- Con WriteBack, una edición en una vista o en el código se escribe en el DataSet: el registro se localiza con Locate sobre el campo Id, se asignan los campos asignados que cambiaron y el registro se confirma de inmediato. Un elemento nuevo añade un registro y un elemento borrado borra su registro. Los recursos nunca se escriben, ni tampoco los campos calculados, de lookup y de solo lectura. No se escribe nada mientras el DataSet está en estado de edición o inserción, y los cambios pendientes se escriben cuando vuelve al estado de exploración.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que se rellenan.
DataSet	TDataSet	Los elementos, un registro por elemento.
ItemFields	TsgcPlannerSourceItemFields	El campo de cada valor del elemento: IdField, TitleField, StartField, FinishField, AllDayField, StatusField, ProgressField, ColorField, ParentField, TagsField, ResourcesField, RankField, NotesField, RecurrenceField y DependsOnField. Los valores por defecto son Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, RecurrenceRule y DependsOn. Un nombre vacío o que no se encuentra en el DataSet se ignora.
ResourceDataSet, ResourceFields	TDataSet, TsgcPlannerSourceResourceFields	Los recursos: IdField, NameField, ColorField e InitialsField, por defecto Id, Name, Color e Initials. Solo lectura.
DependencyDataSet, De-	TDataSet, TsgcPlannerSourceDe-	Los vínculos: FromField, ToField, KindField (FS, SS, FF, SF o el ordinal) y LagField (minutos laborales), por defecto FromId, ToId, Kind y Lag.

pendency-Fields	pendency-Fields	
Active	Boolean	Carga los DataSets abiertos y los sigue. Por defecto False.
LiveUpda-tes	Boolean	Sigue las confirmaciones, inserciones, borrados y refreshes de los DataSets. Por defecto True.
WriteBack	Boolean	Escribe los cambios de los datos en DataSet y DependencyDataSet. Por de-fecto True.

Métodos

Método	Devuelve	Descripción
Load		Carga ahora cada DataSet abierto, con Active o sin él. No ocurre nada cuando Data es nil.
Refresh		Vuelve a cargar cuando Active.

Eventos

Evento	Tipo	Descripción
OnLoa-dItem	TsgcPlanner-SourceLoadIte-mEvent	Después de copiar los campos asignados de un registro a su elemento: aDataSet, en ese registro, y aItem, para que la aplicación pueda rellenar otras propiedades del elemento.

Ejemplo: Tareas y personas desde dos tablas

```
uses
  SysUtils, Classes, DB,
  sgcPlanner_Data, sgcPlanner_Source_DataSet;

type
  TSourceEvents = class
  public
    procedure LoadItem(Sender: TObject; aDataSet: TDataSet;
      aItem: TsgcPlannerItem);
  end;

procedure TSourceEvents.LoadItem(Sender: TObject; aDataSet: TDataSet;
  aItem: TsgcPlannerItem);
begin
  // a column the field maps do not cover
  if aDataSet.FieldByName('Priority').AsInteger > 2 then
    aItem.Tags := 'urgent';
end;

function BindTasks(aOwner: TComponent; aData: TsgcPlannerData;
  aTasks, aPeople: TDataSet; aEvents: TSourceEvents)
  : TsgcPlannerSource_DataSet;
begin
  Result := TsgcPlannerSource_DataSet.Create(aOwner);
  Result.Data := aData;
  // the task table uses its own column names
  Result.DataSet := aTasks;
  Result.ItemFields.IdField := 'TaskId';
  Result.ItemFields.TitleField := 'Name';
```

```
  Result.ItemFields.StartField := 'StartDate';
  Result.ItemFields.FinishField := 'DueDate';
  Result.ItemFields.ResourcesField := 'Owner';
  // the people table matches the default names except one
  Result.ResourceDataSet := aPeople;
  Result.ResourceFields.NameField := 'FullName';
  Result.OnLoadItem := aEvents.LoadItem;
  Result.WriteBack := True;
  Result.Active := True;
end;
```

Consulte también

- TsgcPlannerData
- Inicio rápido VCL
- Vinculación con DataSet

TsgcPlannerICalendar

Unidad: sgcPlanner_ICal
TsgcPlannerICalendar importa y exporta los VEVENT de los archivos iCalendar (RFC 5545), el formato de Google Calendar, Outlook y Apple Calendar, hacia y desde un TsgcPlannerData. Forma parte de la edición Registered (SGC_PLANNER_ADVANCED).

- El Id de un elemento es el UID de su evento. Un evento con un RECURRENCE-ID se convierte en una ocurrencia modificada de su serie.
- Las horas en UTC (Z) y con un TZID se convierten a la hora local, con el VTIMEZONE del archivo. Un TZID desconocido da un aviso y se lee como hora local.
- STATUS se asigna al Status del elemento, RESOURCES a sus recursos (los recursos que faltan se crean), RELATED-TO con RELTYPE=PARENT a ParentId, y RRULE y EXDATE a la recurrencia. Los demás campos del elemento viajan en propiedades X-SGC, de modo que una exportación seguida de una importación da los mismos datos. Las dependencias no se exportan.
- Un texto sin BEGIN:VCALENDAR provoca EsgcPlannerError y deja los datos sin cambios. Una importación es una sola actualización, de modo que las vistas reciben una notificación.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos en los que importar o desde los que exportar.
ImportMode	TsgcPlannerICalImportMode	pimReplace (por defecto) elimina primero los elementos y las dependencias y conserva los recursos, los estados, el calendario y ProjectStart. pimMerge actualiza los elementos con el mismo Id y añade los demás.
ExportUTC	Boolean	Escribe en UTC las horas de los eventos que no se repiten. Las series y las ocurrencias modificadas se escriben siempre como horas locales flotantes, que siguen siendo correctas en los cambios de horario de verano. Por defecto True.
ProductId	string	El PRODID del calendario exportado. Por defecto `-//eSeGeCe//sgcPlanner//EN`.
WarningCount	Integer	Public, solo lectura. Los avisos de la última importación.

Métodos

Método	Devuelve	Descripción
ImportFromFile, ImportFromStream, ImportFromString		Lee un calendario. Los archivos y los streams son UTF-8, con o sin BOM. El UTF-8 no válido se lee como Latin-1, con un aviso.
ExportToFile, ExportToStream		Escribe los datos como un calendario, en UTF-8.
ExportToString	string	Los datos como texto iCalendar.

Eventos

Evento	Tipo	Descripción
OnImport-Warning	TsgcPlannerI-CalWarningE-vent	Una parte del archivo que se omitió o se leyó de forma aproximada: aLine y aMessage, por ejemplo una RRULE fuera del subconjunto admitido, un RDATE o una zona horaria desconocida.

Ejemplo: Combinar un calendario y volver a escribirlo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_ICal;

type
  TICalLog = class
  public
    Lines: TStrings;
    procedure ImportWarning(Sender: TObject; aLine: Integer;
      const aMessage: string);
  end;

procedure TICalLog.ImportWarning(Sender: TObject; aLine: Integer;
  const aMessage: string);
begin
  Lines.Add(Format('line %d: %s', [aLine, aMessage]));
end;

procedure SyncCalendar(aData: TsgcPlannerData; aLog: TICalLog;
  const aFileName: string);
var
  oICal: TsgcPlannerICalendar;
begin
  oICal := TsgcPlannerICalendar.Create(nil);
  try
    oICal.Data := aData;
    // update the events with the same UID, add the new ones
    oICal.ImportMode := pimMerge;
    oICal.OnImportWarning := aLog.ImportWarning;
    oICal.ImportFromFile(aFileName);
    // and write the whole data back as one calendar
    oICal.ExportUTC := True;
    oICal.ExportToFile(ChangeFileExt(aFileName, '.out.ics'));
  finally
    oICal.Free;
  end;
end;
```

Consulte también

- TsgcPlannerMSProject
- TsgcPlannerData
- Ediciones
- iCalendar
- Recurrencia

TsgcPlannerMSProject

Unidad: sgcPlanner_MSProject
TsgcPlannerMSProject importa y exporta archivos XML de Microsoft Project (MSPDI), el formato XML que Microsoft Project guarda y abre, hacia y desde un TsgcPlannerData. Forma parte de la edición Registered (SGC_PLANNER_ADVANCED).

- Las tareas se convierten en elementos: el Id es el UID, el esquema da ParentId, y las fechas, el progreso, el hito, el indicador manual, la restricción y la línea base siguen al archivo. Los vínculos de predecesoras se convierten en dependencias con su tipo y su retraso, y las asignaciones dan los recursos de cada tarea.
- El calendario del proyecto da los días laborables, el horario de trabajo y los festivos, y la fecha de inicio del proyecto da ProjectStart.
- En sgcPlanner solo existen las restricciones ASAP y comienzo no antes de. Debe comenzar el se lee como comienzo no antes de, y los demás tipos de restricción como ASAP, cada uno con un aviso.
- La exportación escribe un calendario Standard, las tareas en el orden del esquema, los vínculos y los recursos. El estado, las etiquetas, el color, el rango, el todo el día y la recurrencia no se escriben.
- Un documento que no está bien formado provoca EsgcPlannerError con la línea y deja los datos sin cambios.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos en los que importar o desde los que exportar.
ImportMode	TsgcPlannerMSProjectImportMode	pmiReplace (por defecto) elimina los elementos y los vínculos y toma el calendario y el inicio del proyecto del archivo. Los recursos y los estados se conservan, y se añaden los recursos del archivo. pmiMerge conserva el calendario y ProjectStart, actualiza los elementos con el mismo Id y añade los demás.
WarningCount	Integer	Public, solo lectura. Los avisos de la última importación.

Métodos

Método	Devuelve	Descripción
ImportFromFile, ImportFromStream, ImportFromString		Lee un proyecto. Los archivos y los streams son UTF-8, con o sin BOM.
ExportToFile, ExportToStream		Escribe los datos como un proyecto.
ExportToString	string	Los datos como texto MSPDI.

Eventos

Evento	Tipo	Descripción
OnImportWarning	TsgcPlannerMSProjectWarningEvent	Una parte del archivo que se leyó de forma aproximada: aLine y aMessage.

Ejemplo: Abrir y guardar un archivo de Microsoft Project

```
uses
  SysUtils, Classes, Dialogs,
  sgcPlanner_Data, sgcPlanner_MSProject;

procedure OpenProject(aData: TsgcPlannerData; const aFileName: string);
var
  oProject: TsgcPlannerMSProject;
begin
  oProject := TsgcPlannerMSProject.Create(nil);
  try
    oProject.Data := aData;
    // tasks, links, calendar and project start from the file
    oProject.ImportMode := pmiReplace;
    oProject.ImportFromFile(aFileName);
    if oProject.WarningCount > 0 then
      ShowMessage(Format('%d warnings', [oProject.WarningCount]));
  finally
    oProject.Free;
  end;
end;

procedure SaveProject(aData: TsgcPlannerData; const aFileName: string);
var
  oProject: TsgcPlannerMSProject;
begin
  oProject := TsgcPlannerMSProject.Create(nil);
  try
    oProject.Data := aData;
    oProject.ExportToFile(aFileName);
  finally
    oProject.Free;
  end;
end;
```

Consulte también

- TsgcVCLGantt
- TsgcPlannerICalendar
- TsgcPlannerData
- XML de MS Project
- Líneas base

TsgcPlannerRenderer

Unidad: sgcPlanner_Renderer
TsgcPlannerRenderer dibuja una vista Scheduler, Gantt o Kanban de un TsgcPlannerData sin ventana, en cualquier TsgcUICanvas, en un PNG o en un SVG, de modo que un servidor, un informe o una aplicación de consola da la misma imagen que muestra un formulario. Posee un modelo de vista de cada tipo, las mismas clases que crean los controles, de modo que la salida coincide píxel a píxel con los controles. Es neutral respecto al framework y se ejecuta en Windows y Linux64, en todas las ediciones.

Propiedades

Propiedad	Tipo	Descripción
Data	TsgcPlannerData	Los datos que se dibujan. La selección es SelectedId de los datos.
View	TsgcPlannerRendererViewKind	prvScheduler (por defecto), prvGantt o prvKanban.
Date	TDateTime	La fecha del Scheduler. 0 (por defecto) sigue el reloj.
Width, Height	Integer	El tamaño en DIP. Por defecto 800 x 600.
Scale	Double	El factor HiDPI. La salida tiene Round(Width * Scale) x Round(Height * Scale) píxeles. Por defecto 1.
Theme, Colors	TsgcUITheme, TsgcPlannerViewColors	Como en los controles.
FontFile, FontFileBold	string	Archivos TrueType con los que medir y dibujar el texto. Vacío (por defecto) usa la fuente incrustada. El Kanban ajusta los títulos de las tarjetas con FontFile.
SchedulerOptions	TsgcPlannerRendererSchedulerOptions	View, Options y TimeScale del Scheduler, como en TsgcVCLScheduler.
GanttOptions	TsgcPlannerRendererGanttOptions	Columns, TreeWidth, RowHeight, TimeScale, DurationUnit, BarColorMode, ShowProgress, ShowToday, ShowDependencies, ShowCriticalPath y ShowBaseline, como en TsgcVCLGantt.
KanbanOptions	TsgcPlannerRendererKanbanOptions	ColumnSource, Columns, ShowUnknownColumn, SwimlaneBy, WIPMode y CardOptions, como en TsgcVCLKanban.
SchedulerView, GanttView, KanbanView		Public, solo lectura. Los modelos de vista, para lo que las opciones no cubren, como ExpandAll, Collapse o ScrollToTime. Su tamaño sigue a Width y Height, de modo que un desplazamiento establecido antes del renderizado se conserva.
ActiveView	TsgcPlannerView	Public, solo lectura. El modelo de vista de View.

Cada clase de opciones también copia las opciones de un modelo de vista, de modo que `GanttOptions.Assign(VCLGantt1.GanttView)` renderiza lo que muestra ese control.

Métodos

Método	Devuelve	Descripción
RenderTo(aCanvas)		Dibuja la vista en un TsgcUICanvas de DeviceWidth x Device-Height píxeles.
SaveToPNG(aFileName or aStream)		La vista como PNG.
SaveToSVG(aFileName or aStream)		La vista como SVG.
DeviceWidth, DeviceHeight	Integer	El tamaño de la salida en píxeles.

Eventos

TsgcPlannerRenderer no tiene eventos. Los eventos OnItemChanging y OnChange de los datos se disparan como siempre cuando los datos cambian.

Ejemplo: Un Gantt como SVG y un mes oscuro como PNG HiDPI

```
uses
  SysUtils, Classes,
  sgcUICore_Theme, sgcPlanner_Data, sgcPlanner_Scheduler, sgcPlanner_Renderer;

procedure ExportViews(aData: TsgcPlannerData; const aFolder: string);
var
  oRenderer: TsgcPlannerRenderer;
begin
  oRenderer := TsgcPlannerRenderer.Create(nil);
  try
    oRenderer.Data := aData;
    oRenderer.Width := 1200;
    oRenderer.Height := 700;
    // the Gantt with every summary open, as SVG
    oRenderer.View := prvGantt;
    oRenderer.GanttOptions.ShowCriticalPath := True;
    oRenderer.GanttView.ExpandAll;
    oRenderer.SaveToSVG(IncludeTrailingPathDelimiter(aFolder) + 'gantt.svg');
    // the month of October in the dark theme, as a HiDPI PNG
    oRenderer.View := prvScheduler;
    oRenderer.Theme := utDark;
    oRenderer.Scale := 2;
    oRenderer.Date := EncodeDate(2026, 10, 1);
    oRenderer.SchedulerOptions.View := psvMonth;
    oRenderer.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'month.png');
  finally
    oRenderer.Free;
  end;
end;
```

Consulte también

- Arquitectura
- TsgcPlannerData
- TsgcVCLGantt
- Renderizado sin ventana

- Temas

Datos compartidos

Cada vista de sgcPlanner muestra un TsgcPlannerData. El objeto de datos contiene todo el planner: los elementos, los recursos, los estados, las dependencias, el calendario laboral, ProjectStart y la selección. Las vistas no guardan copia de él, lo leen, de modo que el Gantt, el tablero Kanban y el Scheduler pueden mostrar los mismos elementos a la vez y un cambio hecho en uno de ellos, o en el código, llega a todos.

Qué contienen los datos

Parte	Contenido
Items	TsgcPlannerItem: Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, Milestone, Manual, Constraint, ConstraintDate, las fechas de línea base y la recurrencia. Un elemento es una tarea en el Gantt, una tarjeta en el tablero y un evento en el Scheduler.
Resources	Personas o salas: Id, Name, Color e Initials. Un elemento nombra sus recursos por Id en su lista Resources separada por comas.
Statuses	El flujo de trabajo: Id, Caption y Color. Son las columnas del tablero y pueden colorear las barras y los eventos.
Dependencies	Los vínculos entre elementos: FromId, ToId, Kind y Lag. Consulte Dependencias.
Calendar	Los días laborables, el horario de trabajo y los festivos. Consulte Calendario laboral.
ProjectStart, SelectedId	El inicio del proyecto para el motor de programación y el elemento seleccionado de cada vista con SyncSelection.

Un elemento se encuentra por su Id con FindItem e IndexOfItem. ParentId construye el árbol de tareas: ChildrenOf('') devuelve las raíces, y un elemento resumen toma sus fechas y su progreso de sus hojas con UpdateSummaries. DeleteItem elimina los vínculos del elemento y sus ocurrencias modificadas, y mueve sus hijos a su padre. MoveItem cambia el estado y el rango de una tarjeta en una sola llamada.

Notificaciones de cambio

Cada setter notifica. Entre BeginUpdate y EndUpdate los cambios se reúnen, y tras el EndUpdate más externo cada listener recibe un TsgcPlannerChange:

Campo	Significado
Kinds	El conjunto de cambios: pckItems (un campo de un elemento), pckStructure (un elemento añadido o borrado, un Id, un ParentId o el orden cambió), pckResources, pckStatuses, pckDependencies, pckCalendar, pckSelection y pckSchedule (ProjectStart, o los resultados TotalSlack y Critical del motor).
ItemId	El Id del elemento cuando cambió exactamente un elemento, si no vacío.
Version	Un número que crece con cada cambio salvo un cambio solo de la selección. Dos versiones iguales significan los mismos datos.

Las vistas son listeners. Un cambio solo de la selección repinta únicamente su overlay, cualquier otro cambio rehace su diseño en el siguiente fotograma. Su propio código escucha con el evento OnChange o con AddListener, que acepta cualquier número de manejadores; llame a RemoveListener antes de liberar el objeto que posee un manejador. Un manejador que vuelve a cambiar los datos inicia una ronda más de notificaciones, y tras 8 rondas se provoca EsgcPlannerError, de modo que dos manejadores que se cambian continuamente entre sí no bloquean la aplicación.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_Gantt_VCL, sgcPlanner_Kanban_VCL;

type
  TChangeLog = class
  public
    Lines: TStrings;
    procedure DataChanged(Sender: TObject; const aChange: TsgcPlannerChange);
  end;

procedure TChangeLog.DataChanged(Sender: TObject;
  const aChange: TsgcPlannerChange);
begin
  // one call per outermost EndUpdate, however many items changed
  if pckStructure in aChange.Kinds then
    Lines.Add('items added or removed')
  else if aChange.ItemId <> '' then
    Lines.Add('item ' + aChange.ItemId + ' changed')
  else if pckItems in aChange.Kinds then
    Lines.Add('several items changed');
end;

procedure StartAll(aData: TsgcPlannerData; aGantt: TsgcVCLGantt;
  aKanban: TsgcVCLKanban; aLog: TChangeLog);
var
  i: Integer;
begin
  // two views, one data object
  aGantt.Data := aData;
  aKanban.Data := aData;
  aData.AddListener(aLog.DataChanged);
  aData.BeginUpdate;
  try
    for i := 0 to aData.Items.Count - 1 do
      if aData.Items[i].Status = 'todo' then
        aData.Items[i].Status := 'doing';
  finally
    // the log, the Gantt and the board get one notification here
    aData.EndUpdate;
  end;
end;

procedure StopLog(aData: TsgcPlannerData; aLog: TChangeLog);
begin
  // remove a listener before the object it belongs to is freed
  aData.RemoveListener(aLog.DataChanged);
end;
```

Consulte también

- Vistas sincronizadas
- TsgcPlannerData
- Arquitectura

Vistas sincronizadas

Vincule dos o más vistas al mismo TsgcPlannerData y se mantienen al unísono. Una tarjeta arrastrada a Done en el tablero se dibuja al instante con el color de Done en el Gantt y en el Scheduler, una barra movida en el Gantt mueve el evento en el Scheduler, y un evento creado en el Scheduler se convierte en una tarjeta. Cada edición se confirma en una sola actualización, de modo que cada vista se repinta una vez por drop.

Selección compartida

Con SyncSelection, que es True por defecto, la selección de una vista es SelectedId de los datos. Haga clic en una barra y se selecciona el mismo elemento en el tablero y en el Scheduler. Con SyncSelection False la vista conserva una selección propia. Establezca SelectedId en el código para seleccionar un elemento en todas las vistas sincronizadas. OnSelectionChange se dispara para una selección hecha en el control o, con SyncSelection, en otra vista de los mismos datos, que es el lugar para desplazar las otras vistas hasta el elemento con ScrollToItem.

Vistas en varios formularios

Ponga el TsgcPlannerData en un módulo de datos y asígnelo a las vistas de cada formulario. Las vistas de un formulario que no está visible también se actualizan y dibujan el estado actual cuando se muestran.

La misma imagen fuera del formulario

Las opciones de una vista se pueden copiar a un TsgcPlannerRenderer con Assign, por ejemplo `Renderer.GanttOptions.Assign(VCLGantt1.GanttView)`, de modo que un informe muestra lo que muestra el formulario. Los controles también se guardan a sí mismos como PNG y SVG. Consulte Renderizado sin ventana.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_Gantt_VCL, sgcPlanner_Scheduler_VCL;

type
  TSyncEvents = class
  public
    Gantt: TsgcVCLGantt;
    Scheduler: TsgcVCLScheduler;
    procedure GanttSelectionChange(Sender: TObject);
  end;

procedure TSyncEvents.GanttSelectionChange(Sender: TObject);
var
  oItem: TsgcPlannerItem;
begin
  // raised for a selection made in any view of the same data
  oItem := Gantt.Data.FindItem(Gantt.Data.SelectedId);
  if oItem = nil then
    Exit;
  Gantt.ScrollToItem(oItem.Id);
  // show the week of the item in the Scheduler, then the item itself
  Scheduler.Date := oItem.Start;
  Scheduler.ScrollToItem(oItem.Id);
end;

procedure SyncViews(aData: TsgcPlannerData; aEvents: TSyncEvents);
begin
  aEvents.Gantt.Data := aData;
```

```
  aEvents.Scheduler.Data := aData;
  aEvents.Gantt.SyncSelection := True;
  aEvents.Scheduler.SyncSelection := True;
  aEvents.Gantt.OnSelectionChange := aEvents.GanttSelectionChange;
  // a selection made in code reaches every view with SyncSelection
  aData.SelectedId := 'design';
end;
```

Consulte también

- Datos compartidos
- Arrastrar y soltar
- Inicio rápido VCL

Calendario laboral

TsgcPlannerData.Calendar es el calendario laboral del planner: los días de la semana que se trabajan, el horario de trabajo de un día y los festivos. Las duraciones, los retrasos, los movimientos del Gantt y el motor de programación cuentan en él minutos laborales, de modo que una tarea de dos días laborables que comienza un viernes termina el lunes por la tarde.

Propiedades

Propiedad	Descripción
WorkDays	Los días de la semana que se trabajan, un conjunto de pwdMonday a pwdSunday. Por defecto de lunes a viernes. Un conjunto vacío significa que todos los días son laborables.
WorkTime	El horario de trabajo de un día laborable como hasta 4 intervalos, `hh:nn-hh:nn` separados por comas. Por defecto `08:00-12:00,13:00-17:00`. Un texto vacío significa todo el día. Un texto no válido provoca EsgcPlannerError, uno válido se almacena en forma canónica.
Holidays	Fechas que no se trabajan, cada una con un Name. AddHoliday(Date, Name) añade una. Un festivo en un día libre se ignora.
FirstDayOfWeek	El primer día de una semana en el Scheduler y en las marcas de semana de la escala de tiempo. Por defecto pwdMonday.

Aritmética del tiempo laboral

Los métodos reciben y devuelven TsgcPlannerTime, el recuento exacto de minutos Int64 del modelo. sgcPlannerTimeOf y sgcPlannerDateTimeOf convierten desde y hacia TDateTime.

Método	Devuelve
IsWorkingDay(aTime), IsWorkingTime(aTime)	Si el día, o el minuto, se trabaja.
WorkingMinutesPerDay	Los minutos laborales de un día laborable, 480 con el WorkTime por defecto.
NextWorkingTime(aTime), PrevWorkingTime(aTime)	El primer minuto laboral en aTime o después, o el último anterior.
AddWorkingMinutes(aTime, aMinutes)	La hora tras aMinutes minutos laborales. Una cantidad positiva devuelve el final del último minuto laboral, de modo que el lunes a las 08:00 más 480 es el lunes a las 17:00, no el martes a las 08:00. Una cantidad negativa retrocede y devuelve el comienzo del último minuto.
WorkingMinutesBetween(aStart, aFinish)	Los minutos laborales de [aStart, aFinish), negativos cuando aFinish es anterior a aStart.
GetNonWorkingIntervals(aStart, aFinish, aList)	Las noches, los días libres y los festivos de un periodo, tal como los dibuja la escala de tiempo.

Dónde se usa el calendario

- El Gantt muestra las duraciones en tiempo laboral (DurationUnit), conserva la duración laboral de una barra que se mueve y sombrea el tiempo no laboral cuando TimeScale.ShowNonWorking es True.
- El Lag de una dependencia es en minutos laborales, y el motor de programación coloca cada tarea en tiempo laboral. Consulte Programación automática.
- UpdateSummaries pondera el progreso de cada hoja por su duración laboral.
- La vista de semana laboral del Scheduler muestra los días de WorkDays.

Un cambio del calendario notifica pckCalendar, de modo que las vistas se repintan, y se guarda con JSON y XML de MS Project.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Const, sgcPlanner_DateUtils, sgcPlanner_Data;

procedure SetupCalendar(aData: TsgcPlannerData);
begin
  // Monday to Thursday, two shifts and the company holidays
  aData.Calendar.WorkDays := [pwdMonday, pwdTuesday, pwdWednesday,
    pwdThursday];
  aData.Calendar.WorkTime := '07:00-13:00,14:00-16:00';
  aData.Calendar.Holidays.AddHoliday(EncodeDate(2026, 12, 8),
    'Immaculate Conception');
  aData.Calendar.Holidays.AddHoliday(EncodeDate(2026, 12, 25), 'Christmas');
end;

function PlanTask(aData: TsgcPlannerData; aItem: TsgcPlannerItem;
  aWorkDays: Integer): Int64;
var
  vStart: TsgcPlannerTime;
begin
  // the first working minute at or after the requested start
  vStart := aData.Calendar.NextWorkingTime(aItem.StartTime);
  aItem.StartTime := vStart;
  // whole working days, the holidays and the days off are skipped:
  // Monday 7 December 07:00 plus 3 days ends on Thursday 10 December 16:00
  aItem.FinishTime := aData.Calendar.AddWorkingMinutes(vStart,
    aWorkDays * aData.Calendar.WorkingMinutesPerDay);
  // 1440 working minutes in the example above
  Result := aData.Calendar.WorkingMinutesBetween(aItem.StartTime,
    aItem.FinishTime);
end;
```

Consulte también

- Programación automática
- Dependencias
- Escala de tiempo y zoom
- TsgcPlannerData

Recurrencia

Edición: Registered. En la edición Community un elemento recurrente se muestra una vez, en su propio Start, y su regla se conserva y se guarda. Consulte Características por edición.
Un elemento con una RecurrenceRule es una serie: se repite en las fechas de una RRULE de iCalendar (RFC 5545). El Scheduler muestra cada ocurrencia, GetOccurrences de los datos las devuelve para cualquier periodo, y los archivos iCalendar las importan y exportan.

La regla

RecurrenceRule contiene el texto de la regla, con o sin el prefijo `RRULE:`. El Start y el Finish del elemento son la primera ocurrencia, y toda ocurrencia tiene la misma duración.

Parte	Compatibilidad
FREQ	DAILY, WEEKLY, MONTHLY o YEARLY.
INTERVAL	Cada n periodos, por ejemplo cada dos semanas.
COUNT, UNTIL	El número de ocurrencias, o la última fecha. UNTIL acepta una fecha, una fecha y hora local o una fecha y hora UTC.
BYDAY	Días de la semana, por ejemplo MO,WE,FR. En las reglas MONTHLY y YEARLY un día puede tener un ordinal de -5 a 5, de modo que 2TU es el segundo martes y -1FR el último viernes. Con DAILY filtra los días.
BYMONTH, BYMONTHDAY	Meses del año y un día del mes (MONTHLY y YEARLY). Una regla YEARLY con BYDAY necesita BYMONTH.
WKST	El primer día de la semana de la regla.

La primera ocurrencia es siempre el Start del elemento y cuenta para COUNT. Una regla no válida hace que el elemento se muestre una vez. Para comprobar una regla escrita por un usuario, llame a TryParse de TsgcPlannerRecurrence, que devuelve False y el motivo.

Excepciones y ocurrencias modificadas

- **RecurrenceExceptions** es una lista separada por comas de fechas o fechas y horas ISO. Una fecha elimina las ocurrencias de todo ese día, una fecha y hora la ocurrencia que comienza en ese minuto. Una ocurrencia eliminada sigue contando para COUNT.
- Una **ocurrencia modificada** es un elemento propio con RecurrenceParentId establecido en el Id de la serie y RecurrenceDate establecido en el inicio de la ocurrencia que sustituye. Puede tener otras horas, otro título u otros recursos, y la ocurrencia generada de esa fecha deja de mostrarse.
- DeleteItem de una serie también elimina sus ocurrencias modificadas.

Edición en el Scheduler

Un drop sobre una ocurrencia generada dispara OnRecurrenceEdit con la serie, la ocurrencia y un ámbito: prsOccurrence (el valor por defecto) crea una ocurrencia modificada, prsSeries mueve toda la serie la misma cantidad y prsCancel cancela el drop. La tecla Supr sobre una ocurrencia dispara OnRecurrenceDelete con los mismos ámbitos: prsOccurrence (el valor por defecto) elimina solo esa ocurrencia, añadiendo su fecha a las excepciones de la serie y borrándola cuando es una ocurrencia modificada, y prsSeries borra toda la serie.
El motor de programación conserva las fechas de una serie y de sus ocurrencias modificadas. El tablero Kanban no muestra las ocurrencias modificadas como tarjetas.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_DateUtils, sgcPlanner_Data, sgcPlanner_Recurrence;

const
  CS_RULE = 'FREQ=WEEKLY;INTERVAL=2;BYDAY=TU;UNTIL=20261231';

function CheckRule(const aRule: string; out aReason: string): Boolean;
var
  oRule: TsgcPlannerRecurrence;
begin
  // validate a rule typed by the user before it is stored
  oRule := TsgcPlannerRecurrence.Create;
  try
    Result := oRule.TryParse(aRule, aReason);
  finally
    oRule.Free;
  end;
end;

procedure AddTeamSync(aData: TsgcPlannerData);
var
  oSeries, oMoved: TsgcPlannerItem;
  vReason: string;
begin
  if not CheckRule(CS_RULE, vReason) then
    raise Exception.Create(vReason);
  aData.BeginUpdate;
  try
    // every second Tuesday at 10:00 until the end of the year
    oSeries := aData.AddItem('sync');
    oSeries.Title := 'Team sync';
    oSeries.Start := EncodeDate(2026, 10, 6) + EncodeTime(10, 0, 0, 0);
    oSeries.Finish := EncodeDate(2026, 10, 6) + EncodeTime(11, 0, 0, 0);
    oSeries.RecurrenceRule := CS_RULE;
    // no meeting on 3 November
    oSeries.RecurrenceExceptions := '2026-11-03';
    // the meeting of 17 November moves to the afternoon
    oMoved := aData.AddItem('sync-1117');
    oMoved.Title := 'Team sync (afternoon)';
    oMoved.RecurrenceParentId := 'sync';
    oMoved.RecurrenceDate := EncodeDate(2026, 11, 17) + EncodeTime(10, 0, 0, 0);
    oMoved.Start := EncodeDate(2026, 11, 17) + EncodeTime(16, 0, 0, 0);
    oMoved.Finish := EncodeDate(2026, 11, 17) + EncodeTime(17, 0, 0, 0);
  finally
    aData.EndUpdate;
  end;
end;

function MeetingsOfQuarter(aData: TsgcPlannerData): Integer;
var
  vList: TsgcPlannerOccurrenceArray;
begin
  // 6: seven Tuesdays, minus 3 November, with 17 November at 16:00
  Result := aData.GetOccurrences(sgcPlannerEncodeDate(2026, 10, 1),
    sgcPlannerEncodeDate(2027, 1, 1), vList);
end;
```

Consulte también

- TsgcVCLScheduler
- iCalendar
- TsgcPlannerData

- Calendario laboral

Escala de tiempo y zoom

El Gantt y la Timeline de recursos del Scheduler dibujan el tiempo en un eje horizontal descrito por un TsgcPlannerTimeScale, la propiedad TimeScale del control (GanttOptions.TimeScale y SchedulerOptions.TimeScale en el renderer). Asigna el tiempo a píxeles, dibuja dos niveles de marcas con etiquetas, sombrea el tiempo no laboral y ajusta los arrastres.

Niveles de zoom

Zoom	Marca menor	Marca mayor	Ancho por defecto	Etiquetas (menor, mayor)
pz5Minutes	5 minutos	hora	20	`nn` , `ddd d mmm t`
pz15Minutes	15 minutos	hora	30	`nn` , `ddd d mmm t`
pzHours	hora	día	40	`hh` , `dddd d mmmm yyyy`
pzDays (por defecto)	día	semana	32	`d` , `d mmm yyyy`
pzWeeks	semana	mes	56	`d mmm` , `mmmm yyyy`
pzMonths	mes	año	60	`mmm` , `yyyy`
pzQuarters	trimestre	año	60	`"Q"q` , `yyyy`
pzYears	año	década	60	`yyyy` , `yyyy`

El eje es lineal en el tiempo. En los zooms gruesos un mes cuenta como 30 días, un trimestre como 91 y un año como 365, de modo que las marcas de meses de distinta duración tienen anchos ligeramente distintos.

Propiedades

Propiedad	Descripción
Zoom	El nivel de zoom. Por defecto pzDays.
TickWidth	El ancho de una marca menor en DIP. 0 (por defecto) es el ancho del zoom en la tabla de arriba.
Origin	El tiempo en X = 0 del eje.
Snap	El paso de un movimiento o un cambio de tamaño: psTick (por defecto, la marca menor hasta pzDays y días completos en zooms más gruesos), psNone, ps5Min, ps15Min, psHour o psDay.
ShowNonWorking	Sombrea las noches, los días libres y los festivos del calendario laboral. Por defecto True.
MajorFormat, MinorFormat	Las etiquetas de los dos niveles. Vacío usa los formatos de la tabla de arriba.

Los tokens de formato no distinguen mayúsculas de minúsculas: `yyyy` y `yy` para el año, `q` para el trimestre, `m` a `mmmm` para el mes, `d` a `dddd` para el día, `w` y `ww` para la semana ISO 8601, `h` y `hh` para la hora de 0 a 23, `n` y `nn` para los minutos y `t` para la hora en el reloj de la configuración regional. El texto entre comillas simples o dobles se copia tal cual. Los nombres proceden de sgcPlannerLocale.

Hacer zoom

- ZoomIn y ZoomOut del Gantt van un nivel más fino o más grueso y devuelven False al final del rango. ZoomToFit elige el zoom que muestra todo el proyecto en el ancho del diagrama.
- Ctrl y la rueda del ratón hacen zoom en el Gantt y en la Timeline en la posición del ratón, de modo que el tiempo bajo el cursor se mantiene en su sitio. En las vistas de día, semana laboral y semana del Scheduler cambian en su lugar la altura del slot.
- ScrollToTime(aTime) desplaza el eje hasta una hora.

En la Timeline de recursos el periodo sigue al zoom: un día hasta pzHours, 4 semanas en pzDays, 3 meses en pzWeeks, un año en pzMonths, 3 años en pzQuarters y una década en pzYears.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_TimeScale, sgcPlanner_Gantt_VCL;

procedure SetupTimeScale(aGantt: TsgcVCLGantt);
begin
  // one minor tick per week, 80 DIPs wide
  aGantt.TimeScale.Zoom := pzWeeks;
  aGantt.TimeScale.TickWidth := 80;
  // the month above, the ISO week number below
  aGantt.TimeScale.MajorFormat := 'mmmm yyyy';
  aGantt.TimeScale.MinorFormat := '"W"ww';
  // a move or a resize snaps to whole days
  aGantt.TimeScale.Snap := psDay;
  aGantt.TimeScale.ShowNonWorking := True;
  aGantt.ScrollToTime(Date);
end;

procedure ZoomClick(aGantt: TsgcVCLGantt; aZoomIn: Boolean);
begin
  // False when the zoom is already at the end of the range
  if aZoomIn then
    aGantt.ZoomIn
  else
    aGantt.ZoomOut;
end;
```

Consulte también

- TsgcVCLGantt
- TsgcVCLScheduler
- Calendario laboral
- Arrastrar y soltar

Arrastrar y soltar

Cada vista edita los datos con el ratón, un lápiz o un dedo. Un arrastre comienza después de que el puntero se mueva 4 DIP, muestra una vista previa mientras dura, desplaza la vista al llegar a un borde y confirma la edición al soltar el botón. Escape o un segundo botón lo cancela.

Los arrastres de cada vista

View	Arrastres
Scheduler	Arrastre sobre un slot vacío para crear un evento (OnItemCreate da el título o lo rechaza). Mueva un evento a otra hora, otro día u otra columna de recurso, y redimensiónelo por su borde superior o inferior. Los movimientos y los cambios de tamaño se ajustan a Options.SlotMinutes. En la banda de todo el día y en la vista de mes se crean y mueven días completos. En la Timeline una barra se mueve a otra hora o a otra fila de recurso y se redimensiona sobre la escala.
Gantt	Mueva una barra, cambie el tamaño de su comienzo o de su final, arrastre el progreso en la mitad inferior de la barra, dibuje un vínculo desde justo fuera del extremo de una barra hasta otra barra (Registered) y arrastre el divisor entre el árbol de tareas y el diagrama. Un movimiento conserva la duración laboral de la tarea.
Kanban	Mueva una tarjeta a otra posición, otra columna u otro carril, o reordene las tarjetas de una columna.

Permitir y rechazar

Ajuste	Efecto
ReadOnly	Desactiva todo arrastre y la tecla Supr del control.
DragOptions	Los arrastres que permite el control: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink y pdoReorder. Por defecto todos. En el tablero, pdoReorder sin pdoMove mantiene cada tarjeta en su columna.
OnItemChanging (datos)	Se dispara una vez por drop, en cualquier vista, con el elemento y los campos que están a punto de cambiar. Ponga aAccept a False para rechazar la edición. Es el único lugar para una regla que debe cumplirse en todas las vistas.
OnCardMoving (Kanban)	Se dispara antes de OnItemChanging con el estado de destino y el carril.
OnItemCreate (Scheduler)	Se dispara para un evento nuevo, con su comienzo, final y recurso.
OnLinkCreating (Gantt)	Se dispara para un vínculo nuevo. Puede cambiar el tipo o rechazar el vínculo.
OnRecurrenceEdit (Scheduler)	Se dispara para un drop sobre una ocurrencia de una serie. Consulte Recurrencia.
WIPMode pwmBlock (Kanban)	Impide que una tarjeta entre en una columna que está en su límite de WIP. Consulte Carriles y límites de WIP.

Qué hace un drop

Un drop es un BeginUpdate y EndUpdate sobre los datos, de modo que cada vista se repinta una vez. Un movimiento o un cambio de tamaño del comienzo en el Gantt también fija la restricción de la tarea en pcSNET en la nueva fecha, salvo que la tarea sea Manual, de modo que una programación posterior la mantiene allí; con AutoSchedule el motor se ejecuta tras el drop. Cuando el elemento tiene un padre, se actualiza su resumen.

Teclado y táctil

La tecla Supr dispara OnItemDelete para el elemento seleccionado y lo borra cuando aAccept sigue en True. En el tablero, SelectNextCard mueve la selección con las flechas, Inicio y Fin. La entrada táctil llega como entrada de ratón, de modo que un dedo arrastra igual que un ratón.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_View, sgcPlanner_Gantt_VCL,
  sgcPlanner_Kanban_VCL, sgcPlanner_Scheduler_VCL;

type
  TDragRules = class
  public
    FrozenUntil: TDateTime;
    procedure ItemChanging(Sender: TObject; aItem: TsgcPlannerItem;
      const aFields: TsgcPlannerItemFields; var aAccept: Boolean);
  end;

procedure TDragRules.ItemChanging(Sender: TObject; aItem: TsgcPlannerItem;
  const aFields: TsgcPlannerItemFields; var aAccept: Boolean);
begin
  // work that started before FrozenUntil keeps its dates, in every view
  if ((pifStart in aFields) or (pifFinish in aFields)) and
    (aItem.Start < FrozenUntil) then
    aAccept := False;
end;

procedure ApplyRole(aData: TsgcPlannerData; aGantt: TsgcVCLGantt;
  aKanban: TsgcVCLKanban; aScheduler: TsgcVCLScheduler; aManager: Boolean;
  aRules: TDragRules);
begin
  aData.OnItemChanging := aRules.ItemChanging;
  if aManager then
  begin
    aGantt.DragOptions := [pdoMove, pdoResize, pdoProgress, pdoLink];
    aKanban.DragOptions := [pdoMove, pdoReorder];
    aScheduler.DragOptions := [pdoCreate, pdoMove, pdoResize];
  end
  else
  begin
    // a team member sees the plan, orders the cards of a column
    // and moves meetings, but creates and resizes nothing
    aGantt.ReadOnly := True;
    aKanban.DragOptions := [pdoReorder];
    aScheduler.DragOptions := [pdoMove];
  end;
end;
```

Consulte también

- Vistas sincronizadas
- TsgcVCLGantt

- TsgcVCLKanban
- TsgcVCLScheduler

Dependencies

Una dependencia une dos elementos: la sucesora (ToId) espera a la predecesora (FromId). Los vínculos se almacenan en Dependencies del TsgcPlannerData en todas las ediciones y se guardan con JSON y XML de MS Project. El Gantt los dibuja como flechas, deja que el usuario los dibuje con el ratón y los usa para programar las tareas en la edición Registered.

Tipos y retraso

Tipo	Significado
pdkFS (por defecto)	Fin a comienzo: la sucesora comienza cuando termina la predecesora.
pdkSS	Comienzo a comienzo: la sucesora comienza cuando comienza la predecesora.
pdkFF	Fin a fin: la sucesora termina cuando termina la predecesora.
pdkSF	Comienzo a fin: la sucesora termina cuando comienza la predecesora.

Lag es en minutos laborales del Calendario laboral, sumados al límite. Un lag negativo es un adelanto, de modo que la sucesora se solapa con la predecesora. AddDependency(FromId, ToId, Kind, Lag) añade un vínculo en una sola llamada.

En el Gantt

- ShowDependencies dibuja cada vínculo como una flecha en codo entre las barras visibles. Los vínculos de la ruta crítica usan el color Critical.
- Arrastre desde justo fuera del comienzo o del final de una barra y suelte sobre otra barra para crear un vínculo. Los extremos elegidos dan el tipo. OnLinkCreating puede cambiar el tipo o rechazar el vínculo. Se rechaza un vínculo al propio elemento, entre un elemento y su ancestro o descendiente, o uno que ya existe.
- Un vínculo en una tarea resumen se aplica a cada tarea hoja que hay debajo cuando se programa el plan.

Mantener los vínculos coherentes

DeleteItem elimina los vínculos del elemento borrado. Un cambio de Id no renombra los vínculos que la usan, así que cambie FromId y ToId de esos vínculos en la misma actualización. Los vínculos con FromId o ToId desconocido se conservan, pero el motor los ignora y no se dibujan. El motor informa de un ciclo de vínculos. Consulte Programación automática.
El origen DataSet lee los vínculos de una tabla propia (DependencyDataSet) o de una lista de predecesoras separadas por comas en la tabla de tareas (DependsOnField). Consulte Vinculación con DataSet.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_View, sgcPlanner_Gantt_VCL;

procedure LinkTasks(aData: TsgcPlannerData);
begin
  aData.BeginUpdate;
  try
    // Build starts when Design finishes
    aData.Dependencies.AddDependency('design', 'build', pdkFS, 0);
```

```
    // Test starts one working day after Build starts
    aData.Dependencies.AddDependency('build', 'test', pdkSS,
      aData.Calendar.WorkingMinutesPerDay);
    // Docs finish two working hours after Test finishes
    aData.Dependencies.AddDependency('test', 'docs', pdkFF, 120);
    // a negative lag is a lead: Review starts 4 working hours
    // before Build finishes
    aData.Dependencies.AddDependency('build', 'review', pdkFS, -240);
  finally
    aData.EndUpdate;
  end;
end;

function Predecessors(aData: TsgcPlannerData; const aId: string;
  aList: TStrings): Integer;
var
  i: Integer;
begin
  aList.Clear;
  for i := 0 to aData.Dependencies.Count - 1 do
    if aData.Dependencies[i].ToId = aId then
      aList.Add(aData.Dependencies[i].FromId);
  Result := aList.Count;
end;

procedure ShowLinks(aGantt: TsgcVCLGantt);
begin
  // Registered edition: arrows between the bars and drag to link
  aGantt.ShowDependencies := True;
  aGantt.DragOptions := aGantt.DragOptions + [pdoLink];
end;
```

Consulte también

- Programación automática
- Ruta crítica
- TsgcVCLGantt
- XML de MS Project

Programación automática

Edición: Registered. En la edición Community, Schedule y SetBaseline del Gantt provocan EsgcPlannerError, AutoSchedule no tiene efecto y la unidad del motor no se incluye. Consulte Características por edición.
El motor de programación, TsgcPlannerScheduleEngine de sgcPlanner_AutoSchedule, coloca las tareas lo antes posible en el calendario laboral después de sus predecesoras, calcula la holgura de cada tarea y marca la ruta crítica. El Gantt lo ejecuta con Schedule, o tras cada edición cuando AutoSchedule es True, y su código puede ejecutarlo sobre un TsgcPlannerData sin ninguna vista, por ejemplo en un servicio.

Las reglas

- Una tarea hoja conserva su duración laboral, los minutos laborales entre su Start y su Finish, y comienza en el primer minuto laboral igual o posterior al mayor de entre el inicio del proyecto, su fecha de restricción y los límites de sus dependencias.
- El inicio del proyecto es ProjectStart de los datos, o el Start más temprano de las tareas hoja cuando ProjectStart no está definido.
- La restricción pcSNET con un ConstraintDate significa comienzo no antes de esa fecha. pcASAP, el valor por defecto, no tiene fecha. Mover una barra en el Gantt establece pcSNET en la nueva fecha, de modo que la siguiente programación mantiene la tarea donde se colocó.
- Una tarea Manual, un elemento recurrente y toda tarea de un ciclo de dependencias conservan sus fechas y siguen condicionando a sus sucesoras.
- Una tarea sin minutos laborales es un hito y queda exactamente en su límite.
- Una tarea resumen toma el inicio más temprano y el final más tardío de sus hojas. Un vínculo en un resumen se aplica a cada una de sus hojas.

Ejecución

Miembro	Descripción
Gantt.Schedule	Ejecuta el motor sobre los datos del Gantt. Devuelve False cuando se encontró un ciclo.
Gantt.AutoSchedule	Ejecuta el motor tras cada edición hecha en el Gantt: un movimiento, un cambio de tamaño y un vínculo nuevo. Los cambios hechos en el código o en otra vista no lo ejecutan, así que llame a Schedule después de ellos.
Gantt.OnScheduleError, Engine.OnScheduleError	Se dispara tras la actualización cuando los vínculos forman un ciclo, con el mensaje y los Ids de las tareas del ciclo. Las demás tareas se programan igualmente.
Engine.Schedule(aData), Engine.CycleIds	La misma pasada sin una vista, y los Ids del último ciclo encontrado.

Todas las fechas y resultados se escriben en una sola actualización: pckItems para las fechas y pckSchedule para la holgura y los indicadores críticos. Una pasada que no cambia nada no notifica nada. En una máquina de desarrollo, 10.000 tareas con 15.000 vínculos y 30 festivos se programan en unos 11 ms (Win64).

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_AutoSchedule;

type
  TScheduleLog = class
  public
```

```
    Errors: TStrings;
    procedure ScheduleError(Sender: TObject; const aError: string;
      aCycleIds: TStrings);
  end;

procedure TScheduleLog.ScheduleError(Sender: TObject; const aError: string;
  aCycleIds: TStrings);
begin
  // the tasks of the cycle keep their dates, the others are scheduled
  Errors.Add(aError);
end;

function ScheduleProject(aData: TsgcPlannerData; aLog: TScheduleLog): Boolean;
var
  oEngine: TsgcPlannerScheduleEngine;
  oItem: TsgcPlannerItem;
begin
  aData.ProjectStart := EncodeDate(2026, 10, 5) + EncodeTime(8, 0, 0, 0);
  // the venue is free from 19 October: Setup starts no earlier
  oItem := aData.FindItem('setup');
  if oItem <> nil then
  begin
    oItem.Constraint := pcSNET;
    oItem.ConstraintDate := EncodeDate(2026, 10, 19) + EncodeTime(8, 0, 0, 0);
  end;
  // the launch date is fixed: the engine never moves a Manual task
  oItem := aData.FindItem('launch');
  if oItem <> nil then
    oItem.Manual := True;
  // a scheduler without any view, for a service or a unit test
  oEngine := TsgcPlannerScheduleEngine.Create;
  try
    oEngine.OnScheduleError := aLog.ScheduleError;
    // False when the links form a cycle
    Result := oEngine.Schedule(aData);
  finally
    oEngine.Free;
  end;
end;
```

Consulte también

- Dependencias
- Ruta crítica
- Líneas base
- Calendario laboral

Ruta crítica

Edición: Registered. En la edición Community, ShowCriticalPath no tiene efecto y el motor que calcula la holgura no se incluye. Consulte Características por edición.
La ruta crítica es la cadena de tareas que fija el final del proyecto: un retraso en cualquiera de ellas retrasa la finalización. El motor de programación la encuentra en cada pasada.

Holgura

- Después de la pasada hacia delante, una pasada hacia atrás desde el final del proyecto, el final más tardío de todas las tareas hoja, da **TotalSlack**: los minutos laborales que una tarea puede retrasarse sin mover el final del proyecto. Es el menor entre la holgura de comienzo y la holgura de final.
- **Critical** es True cuando TotalSlack es 0 o menos. La holgura es negativa cuando una tarea que conserva sus fechas (Manual o recurrente) rompe una dependencia.
- Una tarea resumen toma la menor holgura de sus hojas y es crítica cuando una de ellas lo es. Las tareas de un ciclo no reciben holgura y nunca son críticas.

TotalSlack y Critical son resultados de solo lectura. No se guardan en JSON, así que ejecute Schedule después de cargar un proyecto.

Mostrarla

Ponga ShowCriticalPath del Gantt en True: las barras y los vínculos de las tareas críticas se dibujan con el color Critical del tema, que Colors.Critical sustituye. Ejecute Schedule, o defina AutoSchedule, para que los resultados estén al día.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_AutoSchedule, sgcPlanner_Gantt_VCL;

procedure ListCriticalPath(aData: TsgcPlannerData; aList: TStrings);
var
  i: Integer;
  oItem: TsgcPlannerItem;
  oEngine: TsgcPlannerScheduleEngine;
begin
  // TotalSlack and Critical are results of the engine
  oEngine := TsgcPlannerScheduleEngine.Create;
  try
    oEngine.Schedule(aData);
  finally
    oEngine.Free;
  end;
  aList.Clear;
  for i := 0 to aData.Items.Count - 1 do
  begin
    oItem := aData.Items[i];
    if oItem.IsSummary then
      Continue;
    if oItem.Critical then
      aList.Add(oItem.Title + ': critical')
    else
      // the slack is in working minutes
      aList.Add(oItem.Title + ': ' + IntToStr(oItem.TotalSlack div 60) +
        ' working hours of slack');
  end;
end;
```

```
procedure HighlightCriticalPath(aGantt: TsgcVCLGantt);
begin
  aGantt.ShowCriticalPath := True;
  // the critical bars and links, 0 keeps the colour of the theme
  aGantt.Colors.Critical := $FFC62828;
end;
```

Consulte también

- Programación automática
- Dependencias
- Líneas base
- TsgcVCLGantt

Líneas base

Edición: Registered. En la edición Community, BaselineStart y BaselineFinish se almacenan y se guardan, ShowBaseline no tiene efecto y SetBaseline del Gantt provoca EsgcPlannerError. Consulte Características por edición.
Una línea base es una copia guardada del plan. Cada elemento tiene BaselineStart y BaselineFinish junto a su Start y Finish, y el Gantt dibuja la línea base como una barra fina bajo la barra actual, de modo que una tarea que se retrasó se ve de un vistazo.

Definir la línea base

- SetBaseline del Gantt copia Start y Finish de cada elemento a sus fechas de línea base, en una sola actualización.
- Sin una vista, SetBaseline y ClearBaseline de TsgcPlannerScheduleEngine hacen lo mismo sobre un TsgcPlannerData. ClearBaseline vuelve a poner las fechas de línea base en 0, que significa no definido.
- Las fechas de línea base son propiedades normales, de modo que su código puede definir la línea base de una tarea, o mantener varias líneas base propias y copiar una.

Mostrarla

Ponga ShowBaseline del Gantt en True. Las filas de los elementos con línea base dibujan la barra actual en la parte superior de la fila y la línea base debajo. Compare las fechas en el código con WorkingMinutesBetween del calendario para informar del retraso en tiempo laboral.

Archivos

JSON guarda las fechas de línea base. El XML de MS Project lee y escribe la línea base número 0, e iCalendar las conserva en X-SGC-BASELINE-START y X-SGC-BASELINE-FINISH.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_AutoSchedule, sgcPlanner_Gantt_VCL;

procedure ApprovePlan(aGantt: TsgcVCLGantt);
begin
  // BaselineStart and BaselineFinish of every item take its Start and Finish
  aGantt.SetBaseline;
  aGantt.ShowBaseline := True;
end;

function SlipInWorkingHours(aData: TsgcPlannerData;
  aItem: TsgcPlannerItem): Int64;
begin
  // later than the plan is positive, earlier is negative
  if aItem.BaselineFinishTime = 0 then
    Result := 0
  else
    Result := aData.Calendar.WorkingMinutesBetween(aItem.BaselineFinishTime,
      aItem.FinishTime) div 60;
end;

procedure RebaseWithoutGantt(aData: TsgcPlannerData);
var
  oEngine: TsgcPlannerScheduleEngine;
begin
  oEngine := TsgcPlannerScheduleEngine.Create;
```

```
  try
    // ClearBaseline sets both baseline dates of every item back to 0
    oEngine.ClearBaseline(aData);
    oEngine.Schedule(aData);
    oEngine.SetBaseline(aData);
  finally
    oEngine.Free;
  end;
end;
```

Consulte también

- Programación automática
- Ruta crítica
- TsgcVCLGantt
- XML de MS Project

Carriles y límites de WIP

Edición: Registered. En la edición Community SwimlaneBy actúa como pslNone y WIPLimit no muestra contador y nunca bloquea una tarjeta. Las columnas, los anchos y el plegado funcionan en todas las ediciones. Consulte Características por edición.
El tablero Kanban muestra los elementos como tarjetas en columnas. Los carriles dividen el tablero en bandas horizontales, y los límites de WIP acotan el trabajo en curso de una columna.

Columns

Propiedad	Descripción
ColumnSource	pkcStatuses (por defecto): una columna por estado de los datos, en su orden. pkcList: solo las columnas de Columns.
Columns	Entradas con StatusId, Caption, Color, Width, WIPLimit y Collapsed. Con pkcStatuses una entrada con el StatusId de un estado añade sus ajustes a esa columna. AddColumn(StatusId, Caption, WIPLimit) añade una.
ShowUnknownColumn	Muestra las tarjetas cuyo estado no coincide con ninguna columna en una última columna. No se puede soltar una tarjeta en ella.

Un elemento resumen y una ocurrencia modificada no se muestran como tarjetas. Las tarjetas de una columna se ordenan por Rank, después por su orden en los datos. CollapseColumn y ExpandColumn, o el chevron del encabezado de la columna, pliegan una columna en un raíl estrecho, y OnColumnCollapse lo notifica.

Carriles

SwimlaneBy	Carriles	Un drop en otro carril
pslNone (por defecto)	Sin carriles.	No es posible.
pslResource	Un carril por recurso. Una tarjeta va al carril de su primer recurso.	Sustituye el primer recurso de la tarjeta.
pslParent	Un carril por padre de una tarjeta, en el orden de los datos.	Establece ParentId de la tarjeta.
pslTag	Un carril por primera etiqueta de una tarjeta, ordenados.	No es posible.
pslEvent	Los carriles que OnGetSwimlane devuelve como una clave y un título, por orden de aparición. Sin manejador actúa como pslNone.	No es posible.

Las tarjetas sin carril van a un carril al final.

Límites de WIP

- WIPLimit de una entrada de columna es el mayor número de tarjetas que la columna debería contener, contadas en todos los carriles. 0 significa sin límite.

- El contador del encabezado muestra el recuento, el recuento y el límite por debajo del límite, un contador ámbar en el límite y uno rojo por encima.
- WIPMode pwmWarn (por defecto) solo marca la columna. pwmBlock rechaza una tarjeta que entraría en una columna en su límite, mientras que las tarjetas que ya están en la columna aún se pueden reordenar.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_Kanban, sgcPlanner_Kanban_VCL;

type
  TBoardEvents = class
  public
    procedure GetSwimlane(Sender: TObject; aItem: TsgcPlannerItem;
      var aKey, aCaption: string);
  end;

function HasTag(aItem: TsgcPlannerItem; const aTag: string): Boolean;
var
  i: Integer;
begin
  Result := False;
  for i := 0 to aItem.TagCount - 1 do
    if SameText(aItem.TagAt(i), aTag) then
      Result := True;
end;

procedure TBoardEvents.GetSwimlane(Sender: TObject; aItem: TsgcPlannerItem;
  var aKey, aCaption: string);
begin
  // an expedite lane above the standard work
  if HasTag(aItem, 'urgent') then
  begin
    aKey := 'expedite';
    aCaption := 'Expedite';
  end
  else
  begin
    aKey := 'standard';
    aCaption := 'Standard';
  end;
end;

procedure SetupBoard(aKanban: TsgcVCLKanban; aEvents: TBoardEvents);
begin
  aKanban.ColumnSource := pkcStatuses;
  // In progress holds 2 cards across all lanes, Review 1
  aKanban.Columns.AddColumn('doing', '', 2);
  aKanban.Columns.AddColumn('review', 'Code review', 1);
  // a card dropped on a full column goes back where it was
  aKanban.WIPMode := pwmBlock;
  aKanban.SwimlaneBy := pslEvent;
  aKanban.OnGetSwimlane := aEvents.GetSwimlane;
end;
```

Consulte también

- TsgcVCLKanban
- Arrastrar y soltar
- Temas
- TsgcFMXKanban

Vinculación con DataSet

TsgcPlannerSource_DataSet vincula un TsgcPlannerData a TDataSets: cualquier dataset de la unidad DB, como FireDAC, TClientDataSet, dbExpress o un dataset de terceros. Carga los elementos, los recursos y las dependencias, sigue los cambios de los datasets y escribe de vuelta las ediciones de las vistas. Usa solo la unidad DB, de modo que funciona en VCL, FireMonkey y aplicaciones de consola, en todas las ediciones.

Datasets y mapas de campos

Dataset	Mapa de campos	Nombres de campo por defecto
DataSet	ItemFields	Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, RecurrenceRule y DependsOn.
ResourceDataSet	ResourceFields	Id, Name, Color e Initials.
DependencyDataSet	DependencyFields	FromId, ToId, Kind y Lag.

Un nombre de campo vacío o que no se encuentra en el dataset se ignora, de modo que una tabla asigna solo los campos que tiene. Las fechas se leen de campos de fecha y hora, de campos numéricos o de texto ISO 8601; los booleanos de campos booleanos, números o texto como Y o True; los colores de números ARGB o texto como #RRGGBB. Un Finish nulo da Finish = Start, o la siguiente medianoche para un elemento de todo el día. El tipo (Kind) de una dependencia es FS, SS, FF o SF, y su Lag es en minutos laborales. Sin un DependencyDataSet, el campo DependsOn puede contener los Ids separados por comas de las predecesoras de una tarea, leídos como vínculos de fin a comienzo. OnLoadItem se dispara para cada registro después de copiar los campos asignados, para rellenar otras propiedades del elemento.

Carga y actualizaciones en vivo

Active True carga cada dataset abierto en una sola actualización, de modo que las vistas vinculadas se repintan una vez; 10.000 filas se cargan en unos 26 ms en Win64. Con LiveUpdates, una edición o inserción confirmada actualiza o añade solo ese elemento, y un cambio en el número de registros (un borrado, un filtro o un refresh) sincroniza los elementos por Id. El desplazamiento y las ediciones sin confirmar no cambian nada. Load y Refresh recargan a mano.

Escritura de vuelta

Con WriteBack, True por defecto, una edición de los datos, hecha en una vista o en el código, se confirma en el registro localizado con Locate sobre el campo Id: solo se asignan los campos asignados que cambiaron. Un elemento nuevo añade un registro, un elemento borrado borra su registro, y los vínculos nuevos o eliminados se escriben en el DependencyDataSet o en la lista DependsOn. Mientras el dataset está en estado de edición o inserción no se escribe nada, y los cambios pendientes se escriben cuando vuelve al estado de exploración. Los recursos, los campos calculados, los campos de lookup y los campos de solo lectura nunca se escriben. Una escritura nunca vuelve como una carga nueva.

Ejemplo

```
uses
  SysUtils, Classes, DB, DBClient,
  sgcPlanner_Data, sgcPlanner_Source_DataSet;
```

```
function CreateTaskTable(aOwner: TComponent): TClientDataSet;
begin
  // the default field names: Id, Title, Start, Finish, Status, DependsOn
  Result := TClientDataSet.Create(aOwner);
  Result.FieldDefs.Add('Id', ftString, 20);
  Result.FieldDefs.Add('Title', ftString, 100);
  Result.FieldDefs.Add('Start', ftDateTime);
  Result.FieldDefs.Add('Finish', ftDateTime);
  Result.FieldDefs.Add('Status', ftString, 20);
  Result.FieldDefs.Add('DependsOn', ftString, 200);
  Result.CreateDataSet;
  Result.AppendRecord(['design', 'Design', EncodeDate(2026, 10, 5),
    EncodeDate(2026, 10, 8), 'done', '']);
  Result.AppendRecord(['build', 'Build', EncodeDate(2026, 10, 8),
    EncodeDate(2026, 10, 15), 'doing', 'design']);
end;

procedure BindTable(aSource: TsgcPlannerSource_DataSet;
  aData: TsgcPlannerData; aTasks: TDataSet);
begin
  aSource.Data := aData;
  aSource.DataSet := aTasks;
  // no link table: DependsOn holds the ids of the predecessors (FS links)
  aSource.LiveUpdates := True;
  aSource.WriteBack := True;
  aSource.Active := True;
  // a change of the data is posted to the record at once
  aData.FindItem('build').Status := 'done';
end;
```

La demo DataSetPlanner vincula tres tablas TClientDataSet con sus propios nombres de campo. Consulte Demos.

Consulte también

- TsgcPlannerSource_DataSet
- Datos compartidos
- Persistencia JSON
- Dependencias

Persistencia JSON

Un TsgcPlannerData se guarda en JSON y se carga desde él en todas las ediciones y en todas las plataformas: SaveToJSON y LoadFromJSON para una cadena, SaveToFile y LoadFromFile para un archivo, y las funciones de sgcPlanner_JSON para streams (sgcPlannerJSONSaveToStream y sgcPlannerJSONLoadFromStream) y archivos.

El formato

El documento lleva `"format": "sgcPlanner"` y una versión, después el inicio del proyecto, el calendario, los recursos, los estados, los elementos y las dependencias. Las fechas son horas locales ISO 8601 y los colores son números ARGB. Un proyecto abreviado:

```
{
  "format": "sgcPlanner",
  "version": 1,
  "projectStart": "2026-10-05T08:00:00",
  "calendar": {"workDays":["mon","tue","wed","thu","fri"],"workTime":"08:00-12:00,13:00-17:00","firstDayOfWeek":"mon","holi
  "resources": [
    {"id":"ann","name":"Ann Lee","color":4280973526}
  ],
  "statuses": [
    {"id":"todo","caption":"To do"},
    {"id":"done","caption":"Done","color":4282557941}
  ],
  "items": [
    {"id":"t1","title":"Design","start":"2026-10-05T08:00:00","finish":"2026-10-07T17:00:00","status":"done","progress":100
    {"id":"m1","title":"Weekly review","start":"2026-10-05T09:00:00","finish":"2026-10-05T09:30:00","color":4285887861,"not
  ],
  "dependencies": [
    {"from":"t1","to":"t2","kind":"FS"}
  ]
}
```

Reglas

- Los archivos y los streams son UTF-8 sin marca de orden de bytes. Al cargar se acepta una marca de orden de bytes.
- Un miembro de elemento que contiene su valor por defecto (un texto vacío, false, 0 o una fecha sin definir) no se escribe, salvo id, title, start y finish, lo que mantiene pequeños los archivos.
- Los valores calculados (TotalSlack, Critical, IsSummary), la selección y Version no se guardan.
- Cargar sustituye todos los datos, incluidos el calendario y ProjectStart, en una sola actualización, de modo que cada vista vinculada se repinta una vez. Un miembro ausente carga su valor por defecto.
- Los miembros desconocidos, los miembros de otro tipo JSON y los valores que no se pueden interpretar se ignoran, de modo que un archivo escrito por una versión posterior sigue cargando. Una versión ausente se lee como 1.
- Cuando el texto no es JSON válido, la raíz no es un objeto, el formato es otro o la versión no es compatible, se provoca EsgcPlannerError y los datos quedan como estaban.
- Antes de Delphi 2009 el texto pasa por la página de códigos ANSI del sistema.

En una máquina de desarrollo, 10.000 elementos con 15.000 vínculos se guardan en unos 48 ms y se cargan en unos 56 ms (Win64).

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_DateUtils, sgcPlanner_Data, sgcPlanner_JSON;
```

```
function TryOpenProject(aData: TsgcPlannerData; const aFileName: string;
  out aError: string): Boolean;
begin
  Result := False;
  if not FileExists(aFileName) then
  begin
    aError := 'File not found: ' + aFileName;
    Exit;
  end;
  try
    aData.LoadFromFile(aFileName);
    Result := True;
  except
    on E: EsgcPlannerError do
      // not JSON, not an sgcPlanner document or a newer version:
      // the data is left as it was
      aError := E.Message;
  end;
end;

procedure CopyProject(aSource, aTarget: TsgcPlannerData);
var
  oStream: TMemoryStream;
begin
  oStream := TMemoryStream.Create;
  try
    sgcPlannerJSONSaveToStream(aSource, oStream);
    oStream.Position := 0;
    // replaces everything in aTarget, the calendar included
    sgcPlannerJSONLoadFromStream(aTarget, oStream);
  finally
    oStream.Free;
  end;
end;
```

Consulte también

- TsgcPlannerData
- Vinculación con DataSet
- iCalendar
- XML de MS Project

iCalendar

Edición: Registered. En la edición Community TsgcPlannerICalendar no existe; JSON es el formato de archivo de esa edición. Consulte Características por edición.
TsgcPlannerICalendar importa y exporta calendarios RFC 5545 (.ics), el formato de Google Calendar, Outlook, Apple Calendar y la mayoría de los servidores de calendario. Un VEVENT es un elemento.

Importación

- ImportFromString, ImportFromStream e ImportFromFile leen UTF-8, con o sin marca de orden de bytes. El UTF-8 no válido se lee como Latin-1, con un aviso.
- ImportMode pimReplace (por defecto) elimina primero los elementos y las dependencias y conserva los recursos, los estados, el calendario y ProjectStart. pimMerge actualiza el elemento con el mismo Id y añade los demás.
- Las horas en UTC y las horas con un TZID se convierten a la hora local de la máquina, con el VTIMEZONE del archivo. UTC, GMT y Etc/UTC no necesitan VTIMEZONE. Un TZID desconocido da un aviso y se lee como hora local.
- El Id del elemento es el UID. Un VEVENT con un RECURRENCE-ID se convierte en una ocurrencia modificada de la serie con ese UID.
- Los problemas que no detienen la importación, como una parte de RRULE no admitida o un RDATE, se notifican con OnImportWarning con el número de línea y se cuentan en WarningCount. Un texto sin BEGIN:VCALENDAR provoca EsgcPlannerError y deja los datos sin cambios.

Exportación

- ExportToString, ExportToStream y ExportToFile escriben cada elemento como un VEVENT, con ProductId como PRODID.
- Con ExportUTC (por defecto True) las horas de los elementos que no se repiten se escriben en UTC. Las series y las ocurrencias modificadas usan siempre horas locales flotantes, de modo que sus ocurrencias permanecen a la misma hora en los cambios de horario de verano.
- Un elemento de todo el día cuyo comienzo y final son medianoche se escribe como valores DATE.
- Las dependencias no se exportan.

Correspondencia

iCalendar	Elemento
UID, SUMMARY, DESCRIPTION	Id, Title, Notes.
DTSTART, DTEND	Start, Finish.
RRULE, EXDATE, RECURRENCE-ID	RecurrenceRule, RecurrenceExceptions, RecurrenceParentId y RecurrenceDate.
STATUS	Status en minúsculas: tentative, confirmed o cancelled. Otro estado se conserva en X-SGC-STATUS.
RESOURCES	Los nombres de los recursos. X-SGC-RESOURCES conserva sus Ids cuando difieren, y la importación crea los recursos que faltan.
CATEGORIES	Tags.
RELATED-TO;RELTYPE=PARENT	ParentId.

X-SGC-PROGRESS, X-SGC-COLOR, X-SGC-RANK, X-SGC-MILESTONE, X-SGC-MANUAL, X-SGC-CONSTRAINT, X-SGC-CONSTRAINT-DATE, X-SGC-BASELINE-START, X-SGC-BASELINE-FINISH, X-SGC-ALLDAY	Las propiedades del planner para las que iCalendar no tiene campo, de modo que un archivo del planner hace el viaje de ida y vuelta.

En una máquina de desarrollo, 5.000 VEVENT con zonas horarias se importan en unos 84 ms (Win64). La demo ICalendarSync importa calendarios de Google y Outlook y exporta una serie. Consulte Demos.

Consulte también

- TsgcPlannerICalendar
- Recurrencia
- XML de MS Project
- Persistencia JSON

XML de MS Project

Edición: Registered. En la edición Community TsgcPlannerMSProject no existe; JSON es el formato de archivo de esa edición. Consulte Características por edición.
TsgcPlannerMSProject importa y exporta XML de Microsoft Project (MSPDI), el formato XML que Microsoft Project abre y guarda. Lee tareas, vínculos, el calendario, recursos, asignaciones y líneas base.

Importación

- ImportFromString, ImportFromStream e ImportFromFile leen UTF-8, con o sin marca de orden de bytes. Un documento que no está bien formado provoca EsgcPlannerError con el número de línea.
- ImportMode pmiReplace (por defecto) elimina los elementos y los vínculos y toma el calendario y el inicio del proyecto (StartDate) del archivo. Los recursos y los estados se conservan, y los recursos del archivo se añaden o se renombran.
- pmiMerge conserva el calendario y ProjectStart, actualiza los elementos por Id y añade o actualiza los vínculos por sus dos extremos. Los recursos de los elementos siguen al archivo solo cuando este tiene Assignments.
- OnImportWarning notifica lo que se lee de otra manera, por ejemplo un tipo de restricción que sgcPlanner no tiene.

Correspondencia

MSPDI	sgcPlanner
Task UID, Name, Start, Finish, PercentComplete, Milestone, Notes	Id, Title, Start, Finish, Progress, Milestone, Notes.
OutlineLevel y el orden de las tareas	ParentId, el árbol de tareas.
PredecessorLink Type 0, 1, 2, 3	pdkFF, pdkFS, pdkSF, pdkSS. LinkLag es en décimas de minuto laboral; los formatos de retraso transcurrido y porcentual se convierten.
ConstraintType 0 y 4	pcASAP y pcSNET con ConstraintDate. El tipo 2 (debe comenzar el) se lee como pcSNET y los demás tipos como pcASAP, ambos con un aviso.
Baseline Number 0	BaselineStart y BaselineFinish.
Calendar WeekDays, WorkingTimes, Exceptions	El calendario laboral: días laborables, horario de trabajo y festivos.
Resource Code o Name, Assignments	Los recursos y la lista Resources de cada elemento.

Exportación

ExportToString, ExportToStream y ExportToFile escriben el orden del esquema de SaveVersion 14: un calendario Standard con una excepción por festivo, las tareas en el orden del esquema, sus vínculos y sus asignaciones. Un Id numérico conserva su UID y los demás Ids reciben el siguiente número libre. MSPDI conserva un vínculo por par de tareas. El estado, las etiquetas, el color, el rango, el todo el día y la recurrencia no se escriben, así que mantenga un archivo JSON al lado cuando los necesite.
En una máquina de desarrollo, 10.000 tareas con 15.000 vínculos se importan en unos 264 ms (Win64). La demo ProjectGantt abre y guarda archivos MSPDI. Consulte Demos.

Consulte también

- TsgcPlannerMSProject
- Dependencias
- Líneas base
- Persistencia JSON

Renderizado sin ventana

Cada vista de sgcPlanner se puede dibujar sin ventana. Los modelos de vista dibujan en un TsgcUICanvas de sgcUICore, y los mismos dibujantes sirven a los controles de un formulario, a TsgcPlannerRenderer y a los métodos SaveToPNG y SaveToSVG de los controles, de modo que una imagen tiene los mismos píxeles que la pantalla.

TsgcPlannerRenderer

El renderer es un componente no visual sin unidad de VCL ni de FireMonkey, de modo que se ejecuta en un servicio, en un servidor web, en una aplicación de consola y en Linux64. Funciona en todas las ediciones.

Miembro	Descripción
Data, View	Los datos y la vista que se dibuja: prvScheduler, prvGantt o prvKanban.
Width, Height, Scale	El tamaño en DIP (por defecto 800 x 600) y la escala de la salida, 2 para una imagen HiDPI.
Theme, Colors	Las mismas sustituciones de tema y de colores que en los controles. Consulte Temas.
SchedulerOptions, GanttOptions, KanbanOptions	Las opciones publicadas de cada vista. Assign también acepta un modelo de vista, de modo que `Renderer.GanttOptions.Assign(VCLGantt1.GanttView)` copia lo que muestra un formulario.
Date	La fecha que muestra el Scheduler, 0 para hoy.
SchedulerView, GanttView, KanbanView	Los modelos de vista que posee el renderer, para ExpandAll, ZoomToFit, ScrollToTime o CollapseColumn antes de un renderizado.
RenderTo(aCanvas), SaveToPNG, SaveToSVG	Dibuja en un canvas, o guarda un PNG o un SVG en un archivo o en un stream.
FontFile, FontFileBold	Archivos TrueType para el texto. Vacío usa la fuente incrustada en sgcUICore, que no necesita ninguna fuente instalada en la máquina.

Imágenes de los controles

Los seis controles también tienen SaveToPNG y SaveToSVG, con un nombre de archivo o un stream y ancho, alto y escala opcionales. El tamaño es en DIP y 0 es el tamaño del cliente; una escala de 0 es la escala del control. La salida conserva la posición de desplazamiento del control y es la misma imagen que la de un renderer con los mismos datos, opciones, tamaño y escala. Las barras de desplazamiento, la vista previa del arrastre y las marcas de drop se omiten.

En un servidor

- La selección de la imagen es SelectedId de los datos; bórrela para un informe sin elemento seleccionado.
- Los nombres de días y meses y los formatos de fecha proceden de sgcPlannerLocale, que se inicializa desde la máquina. Defínalo al inicio cuando un servidor deba producir las mismas etiquetas que un escritorio.
- La salida SVG conserva el texto como texto, de modo que se mantiene nítido con cualquier zoom y se puede buscar.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_Gantt, sgcPlanner_Kanban, sgcPlanner_Renderer,
  sgcPlanner_Gantt_VCL;

procedure SaveReport(aData: TsgcPlannerData; const aFolder: string);
var
  oRenderer: TsgcPlannerRenderer;
begin
  // no form and no window: a service, a web server or Linux64
  oRenderer := TsgcPlannerRenderer.Create(nil);
  try
    oRenderer.Data := aData;
    oRenderer.Width := 1200;
    oRenderer.Height := 700;
    // the board with one lane per person
    oRenderer.View := prvKanban;
    oRenderer.KanbanOptions.SwimlaneBy := pslResource;
    oRenderer.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'board.png');
    // the whole plan, every summary open, zoomed to fit the width
    oRenderer.View := prvGantt;
    oRenderer.GanttView.ExpandAll;
    oRenderer.GanttView.ZoomToFit;
    oRenderer.SaveToSVG(IncludeTrailingPathDelimiter(aFolder) + 'plan.svg');
  finally
    oRenderer.Free;
  end;
end;

procedure SaveWhatTheFormShows(aGantt: TsgcVCLGantt);
begin
  // the control itself: same options and scroll position, twice the size
  aGantt.SaveToPNG('gantt.png', 0, 0, 2);
end;
```

La demo HeadlessReport es una aplicación de consola para Windows y Linux64 que escribe cada vista como PNG y SVG. Consulte Demos.

Consulte también

- TsgcPlannerRenderer
- Temas
- Plataformas
- Vistas sincronizadas

Temas

Los controles del planner y el renderer se dibujan con los temas de sgcUICore, de modo que un planner combina con los otros productos sgcUI en el mismo formulario, en VCL, en FireMonkey y en la salida sin ventana.

Tema y colores

- **Theme** es utLight (por defecto), utDark o utCustom. utCustom parte de los colores claros.
- **Colors** sustituye colores sueltos del tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical y Ghost. Un color es un TsgcUIColor ARGB, por ejemplo sgcUIARGB($FF, $00, $96, $88). 0 mantiene el color del tema.

Colores de los elementos

Un elemento se dibuja con el primer color definido: el Color del elemento, el Color de su estado, el Color de su primer recurso y, si no, un color de la paleta del tema elegido a partir de su Id, de modo que el mismo elemento conserva el mismo color en todas las vistas y en todas las ejecuciones.

Ajuste	Efecto
BarColorMode del Gantt	pbcItem (el color del elemento de arriba), pbcStatus (el color del estado) o pbcResource (el color del primer recurso).
Color de las columnas del Kanban	La barra de color del encabezado de una columna. Con pkcStatuses se usa el color del estado cuando no está definido.
Tags	Las píldoras de etiqueta de las tarjetas toman un color de la paleta del tema elegido a partir del texto de la etiqueta.
Resources	Los círculos de iniciales de las tarjetas y de los carriles usan el color del recurso.

Fuentes y tamaños

El texto usa la fuente incrustada en sgcUICore, de modo que la salida es la misma en cualquier máquina. El renderer acepta archivos TrueType con FontFile y FontFileBold. Los tamaños son en DIP y se escalan para el monitor, el formulario o la Scale del renderer.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcUICore_Theme, sgcPlanner_Data, sgcPlanner_Gantt, sgcPlanner_Gantt_VCL,
  sgcPlanner_Kanban_VCL;

procedure ApplyDarkTheme(aGantt: TsgcVCLGantt; aKanban: TsgcVCLKanban);
begin
  aGantt.Theme := utDark;
  aKanban.Theme := utDark;
  // the corporate accent; every other colour comes from the theme
  aGantt.Colors.Accent := sgcUIARGB($FF, $00, $96, $88);
  aKanban.Colors.Accent := sgcUIARGB($FF, $00, $96, $88);
  // 0 hands a colour back to the theme
  aGantt.Colors.NowLine := 0;
end;
```

```
procedure ColourByStatus(aData: TsgcPlannerData; aGantt: TsgcVCLGantt);
var
  oStatus: TsgcPlannerStatus;
begin
  // the bars take the colour of their status
  oStatus := aData.Statuses.Find('done');
  if oStatus <> nil then
    oStatus.Color := sgcUIARGB($FF, $43, $A0, $47);
  aGantt.BarColorMode := pbcStatus;
end;
```

Consulte también

- Renderizado sin ventana
- Carriles y límites de WIP
- TsgcVCLGantt
- TsgcVCLKanban

Características por edición

sgcPlanner se distribuye como Registered, Trial y Community. Trial tiene todas las características de Registered. Esta tabla enumera cada característica y la edición que necesita. Consulte Ediciones para las ediciones en sí.

Característica	Registered y Trial	Community
Datos compartidos, Vistas sincronizadas	Sí	Sí
Calendario laboral	Sí	Sí
Recurrencia	Sí	No. Un elemento recurrente se muestra una vez.
Escala de tiempo y zoom	Sí	Sí
Arrastrar y soltar	Sí	Sí, salvo arrastrar para vincular en el Gantt.
Dependencias	Sí	Los vínculos se almacenan y se guardan, pero no se dibujan.
Programación automática	Sí	No. Schedule provoca EsgcPlannerError.
Ruta crítica	Sí	No.
Líneas base	Sí	Las fechas se almacenan y se guardan, pero no se dibujan. SetBaseline provoca EsgcPlannerError.
Scheduler: vistas de día, semana laboral, semana, mes y agenda	Sí	Sí
Timeline de recursos del Scheduler	Sí	No. La vista muestra un aviso.
Columnas del Kanban, plegado y opciones de tarjeta	Sí	Sí
Carriles y límites de WIP	Sí	No. SwimlaneBy actúa como pslNone.
Vinculación con DataSet	Sí	Sí
Persistencia JSON	Sí	Sí
iCalendar, XML de MS Project	Sí	No. Los componentes no existen.
Renderizado sin ventana, Temas	Sí	Sí

Código que compila en todas las ediciones

Las características avanzadas se compilan bajo SGC_PLANNER_ADVANCED, que sgcVer.inc define en las ediciones Registered y Trial y desactiva en Community. Incluya sgcVer.inc en una unidad que deba compilar con ambas, y ponga el código de las características avanzadas dentro de `{$IFDEF SGC_PLANNER_ADVANCED}`. Las propiedades de las vistas se publican en todas las ediciones, de modo que un formulario que establece ShowCriticalPath o SwimlaneBy carga en Community; un formulario que contiene TsgcPlannerICalendar o TsgcPlannerMSProject no carga.

Ejemplo

```
{$I sgcVer.inc}

uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlanner_Gantt_VCL
{$IFDEF SGC_PLANNER_ADVANCED}
  , sgcPlanner_ICal
{$ENDIF}
  ;

procedure ExportPlan(aData: TsgcPlannerData; const aFileName: string);
{$IFDEF SGC_PLANNER_ADVANCED}
var
  oICal: TsgcPlannerICalendar;
{$ENDIF}
begin
{$IFDEF SGC_PLANNER_ADVANCED}
  oICal := TsgcPlannerICalendar.Create(nil);
  try
    oICal.Data := aData;
    oICal.ExportToFile(ChangeFileExt(aFileName, '.ics'));
  finally
    oICal.Free;
  end;
{$ELSE}
  // Community: JSON is in every edition
  aData.SaveToFile(ChangeFileExt(aFileName, '.json'));
{$ENDIF}
end;

procedure PreparePlan(aGantt: TsgcVCLGantt);
begin
{$IFDEF SGC_PLANNER_ADVANCED}
  // Schedule raises EsgcPlannerError in Community
  aGantt.Schedule;
  aGantt.ShowCriticalPath := True;
{$ENDIF}
  aGantt.ExpandAll;
end;
```

Consulte también

- Ediciones
- Instalación
- Solución de problemas

Sincronización con Google Calendar

TsgcPlannerGoogleSync mantiene un TsgcPlannerData y un Google Calendar en sincronización bidireccional. Este tema explica lo que necesita el componente, cómo se ejecuta una sincronización, cómo se asignan los campos, la regla de conflictos y el archivo de estado.

Requisitos

- La edición Registered de sgcPlanner, con el paquete puente sgcPlannerWS (las unidades sgcPlannerWS_Classes, sgcPlannerWS_Google y sgcPlannerWS_ICal).
- sgcWebSockets, edición Standard o superior, que incluye el cliente de Google Calendar TsgcHTTPGoogleCloud_Calendar_Client. Ponga las fuentes de sgcPlanner, sgcUICore y sgcWebSockets, o la carpeta de la biblioteca de sgcWebSockets, en la ruta de búsqueda.
- Las bibliotecas OpenSSL que sgcWebSockets usa para HTTPS, junto al ejecutable.

El proyecto de Google Cloud y el cliente OAuth2

1. Abra la consola de Google Cloud, cree un proyecto (o elija uno) y active la API de Google Calendar.
2. Configure la pantalla de consentimiento de OAuth. Mientras la aplicación esté en pruebas, añada como usuarios de prueba las cuentas de Google que iniciarán sesión.
3. Cree un ID de cliente OAuth de tipo Aplicación de escritorio y copie su ID de cliente y su secreto de cliente.
4. Configure el cliente: GoogleCloudOptions.Authentication como gcaOAuth2, el ID y el secreto de cliente en GoogleCloudOptions.OAuth2, LocalIP y LocalPort para la redirección del navegador (por ejemplo 127.0.0.1 y 8080), y Scopes.EventsReadWrite y Scopes.CalendarsReadWrite. La sincronización lee el calendario una vez para obtener su zona horaria.
5. En la primera petición el navegador abre la página de consentimiento de Google. Cuando el usuario permite el acceso, la sincronización continúa.

No guarde el secreto de cliente en un archivo de formulario ni en código fuente que distribuya. La demo lo escribe en tiempo de ejecución y lo mantiene solo en memoria.

Cómo se ejecuta una sincronización

1. Hilo de trabajo: la lista de eventos de CalendarId, con el token de sincronización de la última sincronización (solo los cambios desde entonces, incluidos los eventos eliminados), o todos los eventos en la primera sincronización, siguiendo nextPageToken. Las series llegan como un único evento maestro cada una.
2. Hilo principal: los eventos se aplican a Data dentro de un BeginUpdate y EndUpdate. Un evento cancelado elimina su elemento, una ocurrencia cancelada añade una excepción a su serie. Se resuelven los conflictos y la lista de envío se construye con los elementos nuevos, modificados y eliminados, después de OnFilterItem.
3. Hilo de trabajo: inserciones, actualizaciones y eliminaciones en Google.
4. Hilo principal: se guardan los nuevos ID de evento, etags, horas de actualización, huellas y el nuevo token de sincronización, se guarda el estado y se dispara OnSyncDone.

Un elemento leído de Google recibe el id `g:` más el id del evento. Un elemento creado en el planificador conserva su id, que viaja en las propiedades extendidas privadas de su evento (sgcPlannerId), de modo que vuelve con el mismo id.

Asignación de campos

Elemento del planificador	Evento de Google	Notas
Title	summary	

Notas	description	
Start, Finish	start.dateTime, end.dateTime	Se leen con su desfase y se convierten a la hora local. Se escriben en UTC, salvo los eventos recurrentes y sus instancias, que se escriben como hora local con su desfase más timeZone (TimeZone, o la zona horaria del calendario). Las horas tienen precisión de minutos, los segundos se descartan.
AllDay, Start, Finish	start.date, end.date	El final es exclusivo en ambos lados: un elemento de día completo del 1 de octubre va del 1 de octubre 00:00 al 2 de octubre 00:00, y su evento termina el 2 de octubre. Al escribir, un Finish que no está a medianoche se redondea al día siguiente.
RecurrenceRule	the RRULE line of recurrence	Se transmite sin cambios, en ambas direcciones.
RecurrenceExceptions	the EXDATE lines of recurrence	Se escriben como horas UTC, o como fechas en las series de día completo, de modo que no hace falta una base de datos de zonas horarias.
RecurrenceParentId, RecurrenceDate	recurringEventId, originalStartTime	Una ocurrencia modificada de una serie.
Id	extendedProperties.private.sgcPlannerId	El id del elemento en el planificador.

Otras líneas de recurrencia de un evento, como RDATE o EXRULE, se conservan en el estado y se escriben de vuelta sin cambios. Una regla que el motor de recurrencia de sgcPlanner no expande (por ejemplo una con BYSETPOS o HOURLY) también se conserva y se escribe de vuelta sin cambios; el elemento muestra su primera ocurrencia y OnError informa del motivo una vez por regla, de modo que Google nunca pierde una regla.

Qué no se sincroniza

Status, Progress, Color, Tags, Resources y las dependencias de los elementos no se envían a Google, y los asistentes, recordatorios, colores y datos de conferencia de los eventos no se leen. Un elemento editado en el planificador se escribe con el método patch de la API de Calendar, que envía solo los campos que pertenecen al planificador, de modo que los asistentes, recordatorios, datos de conferencia y otras propiedades del evento se conservan.

Conflictos

Un elemento modificado en el planificador y en Google desde la última sincronización es un conflicto. El componente registra cuándo un elemento pasó a ser distinto de su última huella sincronizada; ese momento se compara con la hora `updated` del evento, en UTC:

- Gana el cambio más reciente. Un empate, o un evento sin hora de actualización legible, se resuelve a favor de Google.
- OnConflict recibe el elemento y el JSON del evento y puede asignar a aResolution el valor gcrKeepLocal, gcrKeepRemote o gcrSkip (ambos lados quedan como están en esta ejecución).
- Una edición frente a una eliminación conserva la edición en ambos lados: un evento eliminado se crea de nuevo, o un elemento eliminado se restaura. Cuenta como conflicto pero no dispara OnConflict.
- Un evento cuyo contenido ya coincide con el elemento solo se registra, no se cuenta como conflicto. Los eventos que vuelven del propio envío del componente se reconocen por su etag.
- Con Direction gsdPullOnly los cambios locales nunca cuentan, por lo que Google siempre gana.

Este manejador registra ambos títulos, conserva la versión local de los elementos con la etiqueta `office` y mantiene fuera de Google los elementos con la etiqueta `private`:

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlannerWS_Google;

type
  TSyncRules = class
  public
    Log: TStrings;
    procedure Conflict(Sender: TObject; aItem: TsgcPlannerItem;
      const aEventJSON: string; var aResolution: TsgcPlannerGoogleResolution);
    procedure FilterItem(Sender: TObject; aItem: TsgcPlannerItem;
      var aSync: Boolean);
    procedure Attach(aSync: TsgcPlannerGoogleSync);
  end;

procedure TSyncRules.Attach(aSync: TsgcPlannerGoogleSync);
begin
  aSync.OnConflict := Conflict;
  aSync.OnFilterItem := FilterItem;
end;

procedure TSyncRules.Conflict(Sender: TObject; aItem: TsgcPlannerItem;
  const aEventJSON: string; var aResolution: TsgcPlannerGoogleResolution);
var
  oEvent: TsgcPlannerGoogleEvent;
begin
  oEvent := TsgcPlannerGoogleEvent.Create;
  try
    if oEvent.LoadFromJSON(aEventJSON) then
      Log.Add(Format('conflict: "%s" here, "%s" in Google',
        [aItem.Title, oEvent.Summary]));
  finally
    oEvent.Free;
  end;
  // items tagged "office" are edited here only, keep the local version
  if Pos('office', aItem.Tags) > 0 then
    aResolution := gcrKeepLocal;
  // any other conflict keeps gcrLastModified: the later change wins
end;

procedure TSyncRules.FilterItem(Sender: TObject; aItem: TsgcPlannerItem;
  var aSync: Boolean);
begin
  // private items never leave the application
  if Pos('private', aItem.Tags) > 0 then
    aSync := False;
end;
```

El archivo de estado

El estado es JSON (UTF-8): el nombre y la versión del formato, el id del calendario, el token de sincronización, la zona horaria y, por cada elemento sincronizado, el id local, el id del evento, el etag, la hora `updated`, la huella de los campos sincronizados, el momento en que el elemento cambió y las líneas de recurrencia adicionales. StateFileName lo carga antes de la primera sincronización y lo guarda después de cada una, mediante un archivo temporal que luego se renombra. SaveStateToStream y LoadStateFromStream lo guardan en cualquier otro sitio. El estado de otro calendario se borra cuando cambia CalendarId.
Conserve el estado junto con los elementos que describe. Un elemento sincronizado que falta en Data cuenta como eliminado localmente, y la siguiente sincronización elimina su evento en Google. Cuando los elementos no se guardan, deje StateFileName vacío, de modo que cada inicio comience con una sincronización completa en un planificador vacío. ResetState fuerza una sincronización completa sin eliminar nada.

Errores y sincronizaciones completas

- Un error al listar no cambia nada. Un error al enviar se detiene en la primera llamada que falla. El estado conserva lo que tuvo éxito y el token de sincronización anterior, de modo que la siguiente sincronización lee de nuevo los mismos cambios remotos y reintenta el resto. OnError indica el motivo.
- Cuando Google responde 410 Gone a un token de sincronización caducado, el token se borra y la ejecución se repite como una lista completa. La asignación de los elementos se conserva, de modo que nada se inserta dos veces. Un evento asignado que falta en la lista completa cuenta como eliminado en Google, con la regla de que una edición prevalece sobre una eliminación.
- Una actualización respondida con 404 o 410 se envía de nuevo como inserción. Una eliminación respondida con 404 o 410 cuenta como hecha.

Sincronización con Google y suscripción iCalendar juntas

Un TsgcPlannerICalSubscription puede fusionar un feed de solo lectura en los mismos datos. Sus elementos reciben el IdPrefix `ics:`, que es el ExcludeIdPrefix predeterminado de la sincronización con Google, de modo que los eventos del feed se muestran pero nunca se envían a Google. Si cambia un prefijo, cambie también el otro.

Demos

Las demos de Delphi y C++Builder GoogleCalendarSync e ICalSubscription muestran ambos componentes en un TsgcVCLScheduler. Ejecútelas con `--test` para una autoprueba sin conexión con datos de prueba, sin cuenta de Google. Consulte Demos.

Véase también

- TsgcPlannerGoogleSync
- TsgcPlannerICalSubscription
- Sincronización con Google Calendar e iCalendar en .NET
- Recurrencia
- iCalendar
- TsgcPlannerData

TsgcPlannerGoogleSync

Unidad: sgcPlannerWS_Google
TsgcPlannerGoogleSync mantiene los elementos de un TsgcPlannerData en sincronización bidireccional con un Google Calendar. Forma parte del paquete puente sgcPlannerWS (edición Registered) y llama a la API de Google Calendar mediante el **TsgcHTTPGoogleCloud_Calendar_Client** de sgcWebSockets. El cliente se usa solo como transporte autenticado: el componente llama a sus métodos de nivel API (Event_List, Event_Insert, Event_Patch, Event_Delete y Calendar_Get) y lee y escribe el JSON de los eventos por sí mismo, de modo que el id de calendario `primary` funciona y el OnError del cliente queda para la aplicación. Consulte Sincronización con Google Calendar para la configuración de Google Cloud, la asignación de campos, la regla de conflictos y el archivo de estado.

- Una sincronización lista los eventos modificados (con el token de sincronización guardado, o todos si no hay, siguiendo nextPageToken), los aplica a Data en el hilo principal, después envía los cambios locales (inserciones, actualizaciones y eliminaciones) y guarda en el estado los nuevos ID de evento, etags, huellas y el token de sincronización.
- Los cambios locales se detectan por la huella: se envía un elemento cuyos campos sincronizados difieren de la última sincronización. No hace falta una hora de modificación por elemento.
- Un elemento modificado en ambos lados es un conflicto. Gana el cambio más reciente, un empate se resuelve a favor de Google y OnConflict puede cambiarlo. Una edición siempre prevalece sobre una eliminación.
- Un error HTTP o de autorización detiene la ejecución, conserva el token de sincronización anterior (de modo que la siguiente sincronización reintenta) y dispara OnError.

Propiedades

Propiedad	Tipo	Descripción
Active	Boolean	Por defecto True. Sync y SyncAsync no hacen nada y devuelven False mientras Active es False. La sincronización automática de AutoSyncInterval solo se ejecuta mientras Active.
AutoSyncInterval	Integer	Segundos entre dos sincronizaciones automáticas, 0 (por defecto) las desactiva. Mientras Active, un valor mayor que 0 sincroniza de inmediato en un hilo de trabajo y después cada AutoSyncInterval segundos.
CalendarId	string	El calendario que se sincroniza. Por defecto `primary` , el calendario principal de la cuenta. También funciona el id de otro calendario (se ve en su configuración en Google Calendar).
Client	TsgcHTTPGoogleCloud_Calendar_Client	El cliente de Google Calendar de sgcWebSockets que autentica y envía las peticiones. La aplicación no debe usarlo mientras Busy.
Data	TsgcPlannerData	Los datos del planificador que se sincronizan.
Direction	TsgcPlannerGoogleDirection	gsdTwoWay (por defecto) lee y escribe. gsdPullOnly nunca escribe en Google y los cambios locales nunca cuentan, por lo que Google siempre gana. gsdPushOnly solo envía los cambios locales y no lee los cambios hechos en Google.
ExcludeIdPrefix	string	Los elementos cuyo Id empieza por él nunca se envían. Por defecto `ics:` , el IdPrefix predeterminado

		de TsgcPlannerICalSubscription, de modo que los eventos de una suscripción no van a Google.
StateFileName	string	El archivo JSON del estado de la sincronización. Se carga antes de la primera sincronización y se guarda después de cada una. Vacío (por defecto) mantiene el estado solo en memoria.
TimeZone	string	El nombre de la zona horaria IANA (por ejemplo `Europe/Madrid`) que se escribe con los eventos recurrentes y sus instancias, algo que Google exige. Vacío (por defecto) usa la zona horaria del calendario, leída una vez.
Busy	Boolean	Public, solo lectura. True mientras se ejecuta una sincronización.
LastSyncTime	TDateTime	Public, solo lectura. La hora local al final de la última sincronización correcta.
State	TsgcPlannerGoogleSyncState	Public, solo lectura. El estado de la sincronización: token de sincronización, id del calendario, zona horaria y una entrada por elemento asignado. Cámbielo solo mientras no esté Busy.

Métodos

Método	Devuelve	Descripción
Sync	Boolean	Ejecuta una sincronización en el hilo que llama (el principal) y vuelve cuando termina. False cuando no está Active, cuando está Busy (OnError lo indica) o cuando la ejecución falló (OnError indica el motivo).
SyncAsync	Boolean	Inicia una sincronización en un hilo de trabajo y vuelve de inmediato. Data se modifica y cada evento se dispara en el hilo principal, mediante Synchronize. False cuando no está Active o está Busy.
ResetState		Olvida el token de sincronización y los elementos asignados, de modo que la siguiente sincronización sea completa. Se ignora mientras Busy.
SaveStateToStream		Escribe el estado como JSON (UTF-8), para las aplicaciones que lo guardan con sus propios datos en lugar de usar StateFileName.
LoadStateFromStream	Boolean	Lee un estado escrito por SaveStateToStream o guardado en StateFileName. False, con el estado sin cambios, cuando el texto no es un estado de este formato y versión.

Eventos

Evento	Tipo	Descripción
OnConflict	TsgcPlannerGoogleConflictEvent	Un elemento modificado aquí y en Google desde la última sincronización: aItem es el elemento local y aEventJSON el evento tal como lo devolvió Google. aResolution empieza como gcrLastModified (gana el cambio más reciente, un empate se resuelve a favor de Google). Asigne gcrKeepLocal, gcrKeepRemote o gcrSkip para dejar ambos lados como están en esta ejecución (el elemento si-

		gue modificado). Una edición frente a una eliminación no se notifica aquí, gana la edición.
OnError	TsgcPlannerWSErrorEvent	aError indica por qué falló una sincronización: un estado HTTP, un error de autorización o una sincronización que ya se está ejecutando. También informa, una vez por regla, de una regla de recurrencia que el planificador no puede expandir; esa regla se sigue sincronizando sin cambios.
OnFilterItem	TsgcPlannerGoogleFilterItemEvent	Se llama para cada elemento local antes de enviarlo. Asigne False a aSync para mantener el elemento fuera de Google.
OnSyncDone	TsgcPlannerGoogleSyncDoneEvent	Una sincronización terminó sin error: aPulled eventos aplicados aquí, aPushed elementos escritos en Google, aDeleted eventos eliminados en Google y aConflicts conflictos encontrados.

Hilos

- SyncAsync y la sincronización automática hablan con Google en un hilo de trabajo y modifican Data en el hilo principal, dentro de un BeginUpdate y EndUpdate, de modo que las vistas se repintan una sola vez. Cada evento se dispara en el hilo principal.
- Mientras Busy, solo la sincronización en curso usa Client. No llame al cliente desde la aplicación en ese momento.
- Destruir el componente detiene la sincronización automática y espera a la que esté en curso. No libere el componente dentro de sus propios manejadores de eventos.

Métodos protegidos

El transporte es un conjunto de métodos virtuales protegidos. Cada uno devuelve la respuesta sin procesar y su estado HTTP, y se ejecuta en el hilo de trabajo (o en el hilo principal para Sync). La implementación por defecto llama al cliente sobre CalendarId. Un descendiente puede sobrescribirlos para probar la sincronización sin red, como hacen las demos con su opción `--test`.

Método	Descripción
DoListEvents(const aPageToken, aSyncToken: string; out aStatus: Integer): string	Una página de la lista de eventos.
DoInsertEvent(const aJSON: string; out aStatus: Integer): string	Inserta un evento.
DoUpdateEvent(const aEventId, aJSON: string; out aStatus: Integer): string	Modifica un evento con patch. aJSON contiene solo los campos que pertenecen al planificador, de modo que los demás campos del evento se conservan.
DoDeleteEvent(const aEventId: string; out aStatus: Integer): string	Elimina un evento. 404 y 410 cuentan como hecho.
DoGetCalendar(out aStatus: Integer): string	Lee el calendario, para su zona horaria, cuando ninguna lista de eventos la dio.
DoError(const aError: string)	Dispara OnError.

Ejemplo

Un objeto de sincronización para un formulario o un servicio: OAuth2 con un cliente de aplicación de escritorio, el estado guardado junto al ejecutable, una sincronización automática cada cinco minutos y un registro.

```
uses
  SysUtils, Classes,
  sgcHTTP, sgcHTTP_Google_Cloud, sgcPlanner_Data, sgcPlannerWS_Classes,
  sgcPlannerWS_Google;

type
  TCalendarSync = class
  private
    FClient: TsgcHTTPGoogleCloud_Calendar_Client;
    FLog: TStrings;
    FSync: TsgcPlannerGoogleSync;
    procedure SyncDone(Sender: TObject; aPulled, aPushed, aDeleted,
      aConflicts: Integer);
    procedure SyncError(Sender: TObject; const aError: string);
  public
    constructor Create(aData: TsgcPlannerData; aLog: TStrings;
      const aClientId, aClientSecret: string);
    destructor Destroy; override;
    procedure SyncNow;
  end;

constructor TCalendarSync.Create(aData: TsgcPlannerData; aLog: TStrings;
  const aClientId, aClientSecret: string);
begin
  inherited Create;
  FLog := aLog;
  // the sgcWebSockets client: OAuth2 with a Desktop app client of Google
  FClient := TsgcHTTPGoogleCloud_Calendar_Client.Create(nil);
  FClient.GoogleCloudOptions.Authentication := gcaOAuth2;
  FClient.GoogleCloudOptions.OAuth2.ClientId := aClientId;
  FClient.GoogleCloudOptions.OAuth2.ClientSecret := aClientSecret;
  FClient.GoogleCloudOptions.OAuth2.LocalIP := '127.0.0.1';
  FClient.GoogleCloudOptions.OAuth2.LocalPort := 8080;
  FClient.Scopes.CalendarsReadWrite := True;
  FClient.Scopes.EventsReadWrite := True;

  FSync := TsgcPlannerGoogleSync.Create(nil);
  FSync.Client := FClient;
  FSync.Data := aData;
  FSync.CalendarId := 'primary';
  // keep the sync token and the mapping next to the saved items
  FSync.StateFileName := ExtractFilePath(ParamStr(0)) + 'calendar.state.json';
  FSync.OnSyncDone := SyncDone;
  FSync.OnError := SyncError;
  // sync at once and then every 5 minutes
  FSync.AutoSyncInterval := 300;
end;

destructor TCalendarSync.Destroy;
begin
  // stops the automatic sync and waits for a running one
  FSync.Free;
  FClient.Free;
  inherited;
end;

procedure TCalendarSync.SyncNow;
begin
  if not FSync.SyncAsync then
    FLog.Add('a sync is already running');
end;

procedure TCalendarSync.SyncDone(Sender: TObject; aPulled, aPushed, aDeleted,
  aConflicts: Integer);
begin
  FLog.Add(Format('pulled %d, pushed %d, deleted %d, conflicts %d',
    [aPulled, aPushed, aDeleted, aConflicts]));
end;
```

```
procedure TCalendarSync.SyncError(Sender: TObject; const aError: string);
begin
  // the sync token is kept, the next sync retries
  FLog.Add('sync failed: ' + aError);
end;
```

Véase también

- Sincronización con Google Calendar
- TsgcPlannerICalSubscription
- Sincronización con Google Calendar e iCalendar en .NET
- TsgcPlannerData
- Recurrencia
- Ediciones
- Demos

TsgcPlannerICalSubscription

Unidad: sgcPlannerWS_ICal
TsgcPlannerICalSubscription suscribe un TsgcPlannerData a un feed iCalendar (.ics) de solo lectura publicado en una URL, como la dirección pública o secreta de un calendario de Google, Outlook o Apple, un calendario de equipo o una lista de festivos. Descarga el feed con el cliente HTTP de sgcWebSockets, lo lee con TsgcPlannerICalendar y fusiona sus eventos en los datos. Forma parte del paquete puente sgcPlannerWS (edición Registered, SGC_PLANNER_ADVANCED).

Descarga

- Una URL `webcal://` o `webcals://` se obtiene como `https://`.
- El componente sigue por sí mismo hasta cinco redirecciones (301, 302, 303, 307 y 308), resolviendo los valores relativos de Location contra la URL actual.
- Envía If-None-Match y If-Modified-Since con el ETag y el Last-Modified de la última respuesta importada. 304 Not Modified omite la importación. Un cambio de URL o de IdPrefix olvida estos validadores.
- Un cuerpo mayor que MaxSize, cualquier otro estado, un error de transporte o un feed sin BEGIN:VCALENDAR dispara OnError y deja los elementos y los validadores como estaban, de modo que la siguiente actualización reintenta.

Fusión

- Cada elemento del feed recibe IdPrefix más su id: el UID del evento, o UID#RECURRENCE-ID para una ocurrencia modificada. ParentId y RecurrenceParentId reciben el mismo prefijo.
- Los ids nuevos se añaden, los ids existentes cuyos campos difieren se reemplazan por completo, y los elementos de la suscripción que ya no están en el feed se eliminan, todo dentro de un BeginUpdate y EndUpdate de Data.
- Un elemento pertenece a la suscripción cuando su Id empieza por IdPrefix. Con un IdPrefix vacío, los elementos de la última actualización le pertenecen (se guardan solo en memoria). Los demás elementos de Data nunca se tocan, de modo que los elementos locales y varias suscripciones, cada una con su propio IdPrefix, comparten un mismo objeto de datos.
- Los nombres RESOURCES del feed buscan los recursos de Data, y los recursos nuevos se añaden a Data.
- Las ediciones locales de los elementos de la suscripción se sobrescriben en la siguiente actualización que importe. Vételas en OnItemChanging de Data con IsSubscriptionItem, como muestra el ejemplo.

Propiedades

Propiedad	Tipo	Descripción
Active	Boolean	Por defecto False. True inicia un hilo de trabajo que actualiza de inmediato y después cada RefreshInterval minutos (solo una vez cuando RefreshInterval es 0).
Data	TsgcPlannerData	Los datos del planificador en los que se fusiona el feed.
IdPrefix	string	El prefijo de los ids de los elementos del feed. Por defecto `ics:`, que también es el ExcludeIdPrefix predeterminado de TsgcPlannerGoogleSync, de modo que los eventos del feed nunca se envían a Google.
MaxSize	Integer	El cuerpo más grande que se acepta, en bytes. Por defecto 10485760 (10 MB); 0 o menos significa sin límite.

RefreshInterval	Integer	Minutos entre dos actualizaciones mientras Active, 0 (por defecto) significa una única actualización.
TLSOptions	TsgcHTTPTLS_Options	Las opciones TLS del cliente HTTP propio, para feeds HTTPS.
URL	string	La dirección del feed: https://, http://, webcal:// o webcals://.
Busy	Boolean	Public, solo lectura. True mientras se ejecuta una actualización.
HTTPClient	TsgcHTTP1Client	Public, solo lectura. El cliente HTTP propio de sgcWebSockets, para proxy u otros ajustes. Mientras Busy, solo lo usa la actualización en curso.
LastRefreshTime	TDateTime	Public, solo lectura. La hora local de la última actualización correcta, 0 si no hubo ninguna.

Métodos

Método	Devuelve	Descripción
Refresh	Boolean	Descarga e importa el feed en el hilo que llama (el principal). False cuando está Busy o si hay un error.
RefreshAsync	Boolean	Inicia una actualización en un hilo de trabajo y vuelve de inmediato. La importación y cada evento se ejecutan en el hilo principal. False cuando está Busy.
IsSubscriptionItem(aItem)	Boolean	True cuando aItem pertenece a esta suscripción.

Eventos

Evento	Tipo	Descripción
OnBeforeRefresh	TNotifyEvent	Una actualización empieza, antes de la descarga.
OnError	TsgcPlannerWSErrorEvent	Una actualización falló: aError indica el estado HTTP, el límite de tamaño, demasiadas redirecciones, un error de transporte o un feed que no se puede analizar. Los elementos quedan como estaban.
OnImportWarning	TsgcPlannerICalWarningEvent	Una parte del feed que TsgcPlannerICalendar omitió o leyó de forma aproximada, con aLine y aMessage.
OnRefreshed	TsgcPlannerICalRefreshedEvent	Una actualización tuvo éxito: aAdded, aUpdated y aDeleted elementos. Una respuesta 304 Not Modified da 0, 0 y 0.

Hilos

Refresh bloquea el hilo que llama. RefreshAsync y el bucle de Active descargan en un hilo de trabajo e importan en el hilo principal mediante Synchronize, de modo que cada evento se dispara en el hilo principal. Destruir el componente detiene el hilo de trabajo. No libere el componente dentro de sus propios manejadores de eventos.

Métodos protegidos

Método	Descripción
DoFetch(const aURL: string; aRequestHeaders, aResponseHeaders: TStrings; aBody: TStream): Integer	Un GET de aURL que no sigue redirecciones. aRequestHeaders contiene las líneas de cabecera de la petición (Nombre: valor), aResponseHeaders recibe las líneas de cabecera de la respuesta y aBody el cuerpo. Devuelve el estado HTTP y lanza una excepción ante un error de transporte. Se llama en el hilo de la actualización; las pruebas y las demos lo sobrescriben para leer un archivo local.
DoGetRefreshIntervalMs: Cardinal	El intervalo del bucle de Active en milisegundos.
DoError, DoRefreshed	Disparan OnError y OnRefreshed.

Ejemplo

Un calendario de equipo mostrado junto a los elementos locales, actualizado cada hora, con los eventos del feed de solo lectura en las vistas.

```
uses
  SysUtils, Classes,
  sgcPlanner_Data, sgcPlannerWS_Classes, sgcPlannerWS_ICal;

type
  TTeamFeed = class
  private
    FLog: TStrings;
    FSubscription: TsgcPlannerICalSubscription;
    procedure ItemChanging(Sender: TObject; aItem: TsgcPlannerItem;
      const aFields: TsgcPlannerItemFields; var aAccept: Boolean);
    procedure Refreshed(Sender: TObject; aAdded, aUpdated, aDeleted: Integer);
    procedure FeedError(Sender: TObject; const aError: string);
  public
    constructor Create(aData: TsgcPlannerData; aLog: TStrings;
      const aURL: string);
    destructor Destroy; override;
  end;

constructor TTeamFeed.Create(aData: TsgcPlannerData; aLog: TStrings;
  const aURL: string);
begin
  inherited Create;
  FLog := aLog;
  FSubscription := TsgcPlannerICalSubscription.Create(nil);
  FSubscription.Data := aData;
  FSubscription.URL := aURL; // webcal:// is fetched as https://
  FSubscription.IdPrefix := 'team:';
  FSubscription.RefreshInterval := 60; // minutes
  FSubscription.OnRefreshed := Refreshed;
  FSubscription.OnError := FeedError;
  // feed events are read only in the views
  aData.OnItemChanging := ItemChanging;
  // refresh at once, then every hour
```

```
  FSubscription.Active := True;
end;

destructor TTeamFeed.Destroy;
begin
  FSubscription.Free;
  inherited;
end;

procedure TTeamFeed.ItemChanging(Sender: TObject; aItem: TsgcPlannerItem;
  const aFields: TsgcPlannerItemFields; var aAccept: Boolean);
begin
  if FSubscription.IsSubscriptionItem(aItem) then
    aAccept := False;
end;

procedure TTeamFeed.Refreshed(Sender: TObject;
  aAdded, aUpdated, aDeleted: Integer);
begin
  FLog.Add(Format('added %d, updated %d, deleted %d',
    [aAdded, aUpdated, aDeleted]));
end;

procedure TTeamFeed.FeedError(Sender: TObject; const aError: string);
begin
  // the items of the last good refresh stay in place
  FLog.Add('feed failed: ' + aError);
end;
```

Véase también

- TsgcPlannerICalendar
- TsgcPlannerGoogleSync
- Sincronización con Google Calendar
- iCalendar
- Sincronización con Google Calendar e iCalendar en .NET
- TsgcPlannerData
- Recurrencia
- Ediciones
- Demos

Descripción general de .NET

sgcPlanner .NET es un port administrado en C# de la biblioteca Delphi sgcPlanner, no un envoltorio alrededor de ella. Las clases C# viven en el namespace **esegece.sgcPlanner** y conservan los nombres de Delphi, de modo que **TsgcPlannerData**, **TsgcPlannerItem**, **TsgcPlannerCalendar**, **TsgcPlannerRenderer** y las enumeraciones se leen igual en ambos lenguajes. Las tres vistas siguen el patrón de nombres de Delphi en todos los hosts: **TsgcVCLGantt** es **TsgcWinFormsGantt**, **TsgcWpfGantt** y **TsgcAvaloniaGantt**, y lo mismo el Scheduler y el tablero Kanban. Las vistas se dibujan con el port .NET de **sgcUICore**, el mismo canvas, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe.
Como en Delphi, un único **TsgcPlannerData** contiene los elementos, los recursos, los estados, las dependencias y el calendario laboral, y cualquier número de vistas lo muestra. Una edición en una vista pasa por los datos, que notifican una vez a cada vista vinculada, consulte Datos y programación en .NET.

Paquetes

El port se divide en cinco paquetes NuGet, de modo que un proyecto solo incorpora el host que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcPlanner	Los datos, el calendario laboral, la escala de tiempo, la recurrencia, el motor de programación, los modelos de vista y los painters del Scheduler, de la Timeline de recursos, del Gantt y del tablero Kanban, la fuente **System.Data.DataTable**, JSON, iCalendar y MS Project XML, y la salida SVG de las vistas. Sin dependencia nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPlanner.Rendering	La salida PNG sin ventana: **TsgcPlannerRenderer** y los métodos de extensión **SaveToPNG**, **ExportPNG** y **RenderToRaster** de **sgcPlanner_Raster**, dibujados en el canvas SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPlanner.WinForms	**TsgcWinFormsScheduler**, **TsgcWinFormsGantt** y **TsgcWinFormsKanban** sobre **TsgcPlannerWinFormsView**.	net472, net8.0-windows, net9.0-windows
esegece.sgcPlanner.Wpf	**TsgcWpfScheduler**, **TsgcWpfGantt** y **TsgcWpfKanban** sobre **TsgcPlannerWpfView**.	net472, net8.0-windows, net9.0-windows
esegece.sgcPlanner.Avalonia	**TsgcAvaloniaScheduler**, **TsgcAvaloniaGantt** y **TsgcAvaloniaKanban** sobre **TsgcPlannerAvaloniaView**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0

Los tres paquetes de host dependen de esegece.sgcPlanner.Rendering, que depende de esegece.sgcPlanner, de modo que una línea `PackageReference` trae todo lo que necesita un host. Consulte Instalación desde NuGet. La sincronización con Google Calendar y la suscripción iCalendar del puente sgcWebSockets están en el paquete separado **esegece.sgcPlanner.sgcWebSockets**, solo en la edición Registered.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcPlanner, .Rendering, .WinForms, .Wpf y .Avalonia	Todo: las tres vistas en los tres hosts, arrastrar y soltar, resúmenes, hitos, progreso, temas, la fuente DataTable, JSON, la salida PNG y SVG sin ventana y las características avanzadas: recurrencia, el motor de programación con flechas de dependencia, la ruta crítica y las líneas base, la Timeline de recursos, las swimlanes y los límites de WIP en el tablero, iCalendar y MS Project XML.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcPlanner.WinForms.Community.	Todo excepto las características avanzadas. Las propiedades permanecen, de modo que el mismo código de formulario se compila en ambas ediciones: un elemento recurrente se muestra una vez en su inicio, **ShowDependencies**, **AutoSchedule**, **ShowCriticalPath** y **ShowBaseline** son inertes, **Schedule** y **SetBaseline** del Gantt disparan **EsgcPlannerError**, **psvTimeline** pinta un aviso y **SwimlaneBy** actúa como **psINone**. Las clases de las unidades avanzadas no están en los paquetes Community: **TsgcPlannerScheduleEngine**, **TsgcPlannerICalendar**, **TsgcPlannerMSProject**, **TsgcPlannerSchedulerTimelineView** y las clases de recurrencia.

Es la misma división que las ediciones de Delphi, consulte Ediciones.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (controles TsgcVCL)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	VCL y FMX	Windows, .NET Framework 4.7.2 y .NET 8 y 9. Los controles se pueden declarar en XAML.
Avalonia	FMX (controles TsgcFMX)	Windows, Linux y macOS, .NET 8 y 9.
Sin ventana	TsgcPlannerRenderer y los modelos de vista	Cualquier plataforma de los frameworks de destino: SVG con esegece.sgcPlanner, PNG con esegece.sgcPlanner.Rendering.

Todo host tiene la misma API: las propiedades, los métodos y los eventos de los controles de Delphi con los mismos nombres, más **Data**, **Theme**, **Colors**, **ReadOnly**, **DragOptions**, **SyncSelection**, **ItemAt**, **ScrollToItem**, **SaveToPNG** y **SaveToSVG** en la clase base. Consulte Clases de .NET para los nombres de las clases y Diferencias de .NET respecto a Delphi para lo que cambia respecto a Delphi.

Consulte también

- Instalación desde NuGet
- Inicio rápido WinForms

- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Datos y programación en .NET
- Vinculación de datos en .NET
- JSON, iCalendar y MS Project en .NET
- Renderizado sin ventana en .NET
- Demos de .NET
- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host; los paquetes core y de renderizado vienen con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcPlanner.WinForms
dotnet add package esegece.sgcPlanner.Wpf
dotnet add package esegece.sgcPlanner.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcPlanner.WinForms.Community
dotnet add package esegece.sgcPlanner.Wpf.Community
dotnet add package esegece.sgcPlanner.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcPlanner.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Haga referencia a los paquetes Registered o a los Community en un proyecto, nunca a ambos: contienen los mismos tipos en el mismo namespace.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto .NET Framework 4.7.2. Asigne el namespace en XAML con `xmlns:sgc="clr-namespace:esegece.sgcPlanner;assembly=esegece.sgcPlanner.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Las barras de desplazamiento de las vistas toman su aspecto del tema Fluent. Asigne el namespace en XAML con `xmlns:sgc="using:esegece.sgcPlanner"`.
- Sin ventana: esegece.sgcPlanner solo basta para los datos, los formatos de archivo y SVG. Añada esegece.sgcPlanner.Rendering para PNG, trae SkiaSharp y sus recursos nativos para Windows y Linux.
- Sincronización con Google Calendar y suscripción iCalendar: añada **esegece.sgcPlanner.sgcWebSockets** (edición Registered), que depende de esegece.sgcPlanner y esegece.sgcWebSockets.

En C# todo host necesita `using esegece.sgcPlanner;`, y `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUINotifyEvent** y los demás tipos de sgcUICore.

Consulte también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos de .NET

Inicio rápido WinForms

Este inicio rápido construye un formulario en código con las tres vistas WinForms vinculadas a un **TsgcPlannerData**: un diagrama de Gantt arriba, un tablero Kanban a la izquierda y una semana laboral a la derecha. Arrastre una tarjeta a otra columna y la columna de estado del Gantt la sigue.

Pasos

1. Cree un proyecto Windows Forms para `net8.0-windows` y añada el paquete **esegece.sgcPlanner.WinForms** (**esegece.sgcPlanner.WinForms.Community** en la edición Community).
2. Añada `using esegece.sgcPlanner;` y `using esegece.sgcUICore;`.
3. Cree un **TsgcPlannerData** y rellénelo: estados, recursos y elementos.
4. Cree los controles, asigne **Data** y añádalos al formulario. Son controles WinForms normales: Dock, Anchor, TabIndex y Enabled funcionan como de costumbre.
5. Asigne los eventos. Son los eventos de los controles de Delphi, con los mismos nombres.

Ejemplo

Notas

- El tablero Kanban toma una columna por estado de **Data.Statuses** (**ColumnSource** es **pkcStatuses**). Establezca **ColumnSource** a **pkcList** y rellene **Columns** para elegir las columnas y sus límites de WIP.
- La propiedad **View** del Scheduler es el tipo de vista (**psvDay**, **psvWorkWeek**, **psvWeek**, **psvMonth**, **psvAgenda**, **psvTimeline**), como en Delphi. El modelo de vista es **SchedulerView**; el Gantt y el tablero exponen el suyo como **GanttView** y **KanbanView**.
- **Theme** es por control. Las barras de desplazamiento son las nativas de Windows, y **BorderStyle** elige el borde.
- **SaveToPNG** y **SaveToSVG** escriben lo que muestra el control, a su tamaño o a otro tamaño y escala.

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Datos y programación en .NET
- Inicio rápido VCL

Inicio rápido WPF

Los controles WPF se pueden declarar en XAML como cualquier otro control. Este inicio rápido declara un Gantt, un tablero Kanban y un Scheduler en un grid y los vincula a un **TsgcPlannerData** en el code-behind.

Pasos

1. Cree un proyecto WPF para `net8.0-windows` y añada el paquete **esegece.sgcPlanner.Wpf** (**esegece.sgcPlanner.Wpf.Community** en la edición Community).
2. Asigne el namespace en la ventana: `xmlns:sgc="clr-namespace:esegece.sgcPlanner;assembly=esegece.sgcPlanner.Wpf"`.
3. Declare los controles. Las propiedades simples se pueden establecer como atributos, y las enumeraciones toman los nombres de sus valores, como `View="psvWorkWeek"`. Un control sin Width y Height en un panel que no lo estira pide 400 x 300.
4. Cree los datos en el code-behind y asígnelos a **Data** de cada control.

Ejemplo

La ventana:

```
<Window x:Class="QuickStart.QuickStartWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:sgc="clr-namespace:esegece.sgcPlanner;assembly=esegece.sgcPlanner.Wpf"
        Title="sgcPlanner" Width="1100" Height="700">
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="260" />
      <RowDefinition Height="*" />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="560" />
      <ColumnDefinition Width="*" />
    </Grid.ColumnDefinitions>
    <sgc:TsgcWpfGantt x:Name="ganTasks" Grid.ColumnSpan="2" TreeWidth="260" ShowToday="False" />
    <sgc:TsgcWpfKanban x:Name="kbnBoard" Grid.Row="1" />
    <sgc:TsgcWpfScheduler x:Name="schWeek" Grid.Row="1" Grid.Column="1" View="psvWorkWeek" />
  </Grid>
</Window>
```

El code-behind:

Notas

- Las propiedades son propiedades CLR simples, no propiedades de dependencia, así que admiten valores literales en XAML pero no `{Binding}`. Los datos de un plan se vinculan mediante **TsgcPlannerData**, consulte Vinculación de datos en .NET.
- Las barras de desplazamiento son hijos `ScrollBar` de WPF del control. **IsTabStop** ocupa el lugar de **TabStop** de WinForms, y no hay **BorderStyle**: use BorderBrush y BorderThickness o un Border.
- Los controles ejecutan un temporizador para la línea de la hora actual mientras están cargados. **Dispose** detiene los temporizadores y desasocia el listener de los datos; llámelo al cerrar la ventana.

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET

- Vinculación de datos en .NET
- TsgcVCLGantt

Inicio rápido Avalonia

Los controles Avalonia se ejecutan en Windows, Linux y macOS. Este inicio rápido es una aplicación Avalonia completa en código: un tablero Kanban junto a una semana agrupada por recurso, ambos en el tema oscuro y vinculados a un **TsgcPlannerData**.

Pasos

1. Cree un proyecto de consola `net8.0`, establezca `<OutputType>WinExe</OutputType>` y añada **Avalonia.Desktop** y **Avalonia.Themes.Fluent** 11.3 y el paquete **esegece.sgcPlanner.Avalonia** (**esegece.sgcPlanner.Avalonia.Community** en la edición Community).
2. Añada el tema Fluent a la aplicación.
3. Cree los datos y los controles y añada los controles a un panel. En XAML asigne el namespace con `xmlns:sgc="using:esegece.sgcPlanner"`.

Ejemplo

Notas

- **Theme** de los controles es el tema de sgcUICore. Oculta el **StyledElement.Theme** de Avalonia, así que establézcalo mediante una variable del tipo del control, como en el ejemplo.
- **GroupByResource** da a cada persona un grupo de columnas en las vistas de día, semana laboral y semana, en todas las ediciones. La Timeline de recursos (**psvTimeline**) es una característica de Registered.
- Las propiedades son propiedades CLR simples, no propiedades estilizadas: establézcalas en código o como atributos XAML.
- En macOS la tecla Comando cuenta como Ctrl, así que Comando y la rueda hacen zoom en el Gantt.

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Datos y programación en .NET
- Inicio rápido FireMonkey

Clases de .NET

Cada clase de Delphi conserva su nombre en .NET. Los controles VCL y FireMonkey pasan a ser los controles WinForms, WPF y Avalonia con el prefijo del host, y las clases independientes del framework son las mismas en todos los hosts.

Clases de host

Delphi	WinForms	WPF	Avalonia	Modelo (propiedad del host)
TsgcVCLScheduler, TsgcFMXScheduler	TsgcWinFormsScheduler	TsgcWpfScheduler	TsgcAvaloniaScheduler	TsgcPlannerSchedulerView (SchedulerView): TsgcPlannerSchedulerTimelineView en la edición Registered, TsgcPlannerSchedulerRenderView en la edición Community
TsgcVCLGantt, TsgcFMXGantt	TsgcWinFormsGantt	TsgcWpfGantt	TsgcAvaloniaGantt	TsgcPlannerGanttRenderView (GanttView)
TsgcVCLKanban, TsgcFMXKanban	TsgcWinFormsKanban	TsgcWpfKanban	TsgcAvaloniaKanban	TsgcPlannerKanbanRenderView (KanbanView)
TsgcPlannerVCLView, TsgcPlannerFMXView	TsgcPlannerWinFormsView	TsgcPlannerWpfView	TsgcPlanneRAvaloniaView	TsgcPlannerView (View)

Las clases base derivan de **TsgcUIWinFormsControl**, **TsgcUIWpfControl** y **TsgcUIAvaloniaControl** de sgcUICore.

Miembros de cada control

Miembro	Descripción
Data	El **TsgcPlannerData** que muestra el control. Varios controles pueden compartir uno.
View	El modelo de vista, un **TsgcPlannerView**. Las propiedades tipadas SchedulerView, GanttView y KanbanView devuelven el mismo objeto. En el Scheduler, **View** se redeclara como el tipo de vista, como en Delphi.
Theme, Colors	El aspecto: el **TsgcUITheme** (utLight, utDark) y los **TsgcPlannerViewColors** que sustituyen los colores del tema.
ReadOnly, DragOptions	Desactivan la edición, o eligen los arrastres que puede hacer el usuario (crear, mover, redimensionar, progreso, vincular, reordenar), como `[Flags]` **TsgcPlannerDragOptions**.
SyncSelection	La selección sigue a **SelectedId** de los datos, de modo que las vistas vinculadas a los mismos datos seleccionan el mismo elemento.
ItemAt, ScrollToItem	El elemento en un punto del control (píxeles de cliente en WinForms, DIP en WPF y Avalonia) y el desplazamiento de un elemento hasta hacerlo visible.

UpdateScrollBars, CursorKind	Actualizan las barras de desplazamiento tras un cambio hecho en el modelo, y el cursor que muestra el control.
OnItemClick, OnItemDblClick, OnItemDelete, OnSelectionChange	Los eventos de la vista base de Delphi. OnItemDelete puede rechazar el borrado.
SaveToPNG, SaveToSVG	Dibujan la vista tal como está ahora en un stream o en un archivo, al tamaño del control o a otro tamaño, con una escala.

Tipos independientes del framework

Tipo	Descripción
TsgcPlannerData	Elementos, recursos, estados, dependencias, el calendario laboral, la notificación de cambios, JSON. Consulte TsgcPlannerData.
TsgcPlannerItem, TsgcPlannerResource, TsgcPlannerStatus, TsgcPlannerDependency	Los elementos de las colecciones, en **TsgcPlannerItems**, **TsgcPlannerResources**, **TsgcPlannerStatuses** y **TsgcPlannerDependencies**.
TsgcPlannerCalendar, TsgcPlannerHolidays	Los días laborables, los horarios de trabajo y los festivos. Consulte Calendario laboral.
TsgcPlannerTimeScale	El zoom y el ajuste del Gantt y de la Timeline. Consulte Escala de tiempo.
TsgcPlannerSchedulerView, TsgcPlannerGanttView, TsgcPlannerKanbanView	Los modelos de vista con sus subclases de renderizado: diseño, detección de aciertos, teclado, ratón y arrastre, y **ExportImage** para SVG. No necesitan ventana.
TsgcPlannerScheduleEngine	El motor de programación (Registered). Consulte Programación automática.
TsgcPlannerSource_DataSet	Vincula los datos a tablas **System.Data.DataTable**. Consulte Vinculación de datos en .NET.
TsgcPlannerICalendar, TsgcPlannerMSProject	Importación y exportación de iCalendar y MS Project XML (Registered). Consulte JSON, iCalendar y MS Project en .NET.
TsgcPlannerRenderer	En esegece.sgcPlanner.Rendering: las tres vistas a PNG y SVG sin ventana. Consulte Renderizado sin ventana en .NET.
TsgcPlannerLocale	Los nombres de los días y de los meses, el formato de hora y el separador decimal de las etiquetas de cabecera, tomados de `CultureInfo.CurrentCulture`.
sgcPlanner_DateUtils, sgcPlanner_JSON, sgcPlanner_Raster	Clases estáticas con las funciones libres de Delphi: las conversiones de tiempo y el reloj, las funciones JSON y los métodos de extensión PNG de las vistas en esegece.sgcPlanner.Rendering.

Enumeraciones

Las enumeraciones conservan los nombres y los prefijos de valor de Delphi, por ejemplo **TsgcPlannerSchedulerViewKind** (psv), **TsgcPlannerGanttField** (pgc), **TsgcPlannerBarColorMode** (pbc), **TsgcPlannerDurationUnit** (pdu), **TsgcPlannerDependencyKind** (pdk), **TsgcPlannerConstraint** (pc), **TsgcPlannerKanbanColumnSource**

(pkc), **TsgcPlannerSwimlaneBy** (psl), **TsgcPlannerWIPMode** (pwm), **TsgcPlannerRecurrenceScope** (prs) y **TsgcPlannerRendererViewKind** (prv). Los sets de Delphi son enumeraciones `[Flags]` combinadas con `|`: **TsgcPlannerWeekDays**, **TsgcPlannerChangeKinds**, **TsgcPlannerItemFields** y **TsgcPlannerDragOptions**. El **TAlignment** de la RTL de Delphi de las columnas del Gantt se declara en esegece.sgcPlanner.

Consulte también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Arquitectura

Diferencias de .NET respecto a Delphi

El port .NET conserva los nombres, los valores por defecto, el comportamiento del teclado y del ratón y las reglas de diseño de Delphi. La salida SVG de las vistas es idéntica byte a byte a la de Delphi, y la salida PNG coincide con ella dentro de la tolerancia del rasterizador SkiaSharp. Las diferencias siguientes provienen de los frameworks de los hosts y de la biblioteca de clases de .NET.

Tiempos y valores

- Las propiedades de fecha son valores `DateTime` de Kind Unspecified: **Start**, **Finish**, **ConstraintDate**, **BaselineStart**, **BaselineFinish**, **RecurrenceDate** y **ProjectStart**. El TsgcPlannerTime de Delphi, en minutos enteros, es un `long`, al que se accede por las propiedades con el sufijo Time: **StartTime**, **FinishTime** y así sucesivamente. **Duration** está en minutos.
- El reloj es `sgcPlanner_DateUtils.sgcPlannerNow`, un `Func<DateTime>`. Sustitúyalo para congelar el tiempo en las pruebas y restaure después el delegado guardado.
- TsgcUIColor es un `uint`, $AARRGGBB, donde 0 significa el color del tema o del estado.
- TFormatSettings y los globales de la RTL se sustituyen por `CultureInfo.CurrentCulture`, capturada por **TsgcPlannerLocale**. Las propiedades indexadas de los nombres de días y meses son los métodos **GetShortDayName**, **SetShortDayName**, **GetLongMonthName** y así sucesivamente. Los archivos se escriben siempre con formatos invariantes.

Eventos, métodos y listas

- Los eventos son propiedades delegado con los mismos nombres que en Delphi (**OnChange**, **OnItemCreate**, **OnCardMoving**...). Asigne un manejador con `=`. Un parámetro `var` de Delphi es un parámetro `ref`, y una lambda con un parámetro `ref` declara el tipo de cada parámetro. **TNotifyEvent** es **TsgcUINotifyEvent**.
- Los datos aceptan varios listeners: **AddListener** y **RemoveListener** reciben un **TsgcPlannerChangeEvent**, como las vistas.
- Una propiedad indexada de Delphi es un método: `Rows(i)`, `ColumnInfo(i)`, `Cards(i)`, `Entries(i)`, `Calendar.Intervals(i)`. Una función de Delphi sin parámetros sigue siendo un método, como `RowCount()` o `WorkingMinutesPerDay()`.
- Un array `var` de Delphi es un array `ref` que el método sustituye por uno de la longitud exacta: `ChildrenOf(id, ref aList)`, `GetOccurrences(start, finish, ref aList)`.
- Un set de Delphi es una enumeración `[Flags]` combinada con `|`. Las colecciones conservan la forma de TCollection: Add, Delete, Clear, Assign, BeginUpdate, EndUpdate, un indexador y `foreach`.

Sin streaming de componentes ni notificación de liberación

- No hay DFM en .NET. Los controles se crean en código o en XAML, y los datos, las columnas y las opciones se establecen en código.
- .NET no tiene notificación de liberación. Una vista conserva sus datos hasta que **Data** se establece a null o se libera el control. Los controles WPF y Avalonia y los modelos de vista son `IDisposable`: **Dispose** detiene sus temporizadores y desasocia el listener de los datos. Avalonia no libera los controles, así que libérelos al cerrar la ventana.
- Los datos no tienen estado de carga: los cambios se acumulan entre **BeginUpdate** y **EndUpdate** y se notifican una vez.

Hosts

- WinForms usa las barras de desplazamiento nativas y el **BorderStyle** de WinForms; WPF y Avalonia tienen hijos `ScrollBar` y no tienen BorderStyle.
- WPF y Avalonia tienen **IsTabStop** en lugar de **TabStop**. En Avalonia la tecla Comando cuenta como Ctrl, y **Theme** oculta el **StyledElement.Theme** de Avalonia.
- Las propiedades de los controles son propiedades CLR simples en todos los hosts, no propiedades de dependencia ni estilizadas.
- El toque llega como entrada de ratón; el arrastre táctil de Delphi no tiene una ruta de código aparte.

- El cursor de división (**pcrSplit**) es un cursor de redimensionado horizontal en WPF y Avalonia, que no tienen cursor de división.

Fuentes de datos y archivos

- **TsgcPlannerSource_DataSet** vincula tablas `System.Data.DataTable` en lugar de TDataSet. Consulte Vinculación de datos en .NET para ver las diferencias.
- MS Project XML se lee con `System.Xml`. La unidad sgcPlanner_XML de Delphi no se ha portado y no hace falta.
- **ExportImage** de un modelo de vista escribe solo SVG. El PNG necesita esegece.sgcPlanner.Rendering, consulte Renderizado sin ventana en .NET.

Consulte también

- Descripción general de .NET
- Clases de .NET
- Datos y programación en .NET
- Vinculación de datos en .NET

Datos y programación en .NET

TsgcPlannerData es la única fuente de verdad de un plan. Contiene los elementos (tareas, eventos e hitos), los recursos, los estados, las dependencias y el calendario laboral. Las vistas nunca escriben por sí mismas un campo de un elemento: cada soltar llama a los datos dentro de un único **BeginUpdate** y **EndUpdate**, y los datos notifican una vez a cada listener, de modo que un Gantt, un tablero y un Scheduler vinculados a los mismos datos siempre coinciden.

Construir un plan

El ejemplo construye un pequeño proyecto con tres estados, dos personas, un calendario con dos turnos y un festivo, cuatro tareas enlazadas de fin a inicio y un hito, luego escucha los cambios, mueve una tarjeta como lo hace un soltar de Kanban y ejecuta el motor de programación.

Notas

- **Finish** es exclusivo, también para los elementos de día completo: un elemento de día completo el 1 de octubre va del 1 de octubre a las 00:00 al 2 de octubre a las 00:00. Un elemento con **Start** igual a **Finish** es un hito.
- **Resources** y **Tags** son listas separadas por comas; **ResourceAt**, **ResourceCount**, **TagAt** y **TagCount** las leen una a una. **ParentId** construye el árbol de tareas y **UpdateSummaries** recalcula las tareas de resumen.
- **MoveItem** cambia el estado y el orden de una tarjeta, **DeleteItem** elimina un elemento y los elementos de excepción de su serie, y **SelectedId** es la selección compartida por las vistas.
- **OnItemChanging** puede rechazar una edición antes de que se haga, para los campos de un set **TsgcPlannerItemFields**.
- El motor coloca cada tarea que no es **Manual** lo antes posible tras sus vínculos y su **Constraint**, en el calendario laboral, y calcula **TotalSlack** y **Critical**. Un ciclo de vínculos dispara **OnScheduleError** con los ids del ciclo. Un Gantt con **AutoSchedule** lo ejecuta tras cada edición, **Schedule** del Gantt lo ejecuta a petición, y **SetBaseline** guarda las fechas actuales como línea base. El motor es una característica de Registered.

Consulte también

- Descripción general de .NET
- Clases de .NET
- JSON, iCalendar y MS Project en .NET
- Datos compartidos
- Programación automática
- Calendario laboral

Vinculación de datos en .NET

El **TsgcPlannerSource_DataSet** de Delphi carga un plan desde tablas TDataSet y escribe de vuelta las ediciones. En .NET la misma clase vincula tablas `System.Data.DataTable`, las tablas que rellena un `DataAdapter` de cualquier proveedor ADO.NET. Las vistas siguen vinculadas al **TsgcPlannerData**, de modo que la misma fuente sirve a la vez a un Gantt, a un tablero y a un Scheduler.

Propiedad	Descripción
Data	El **TsgcPlannerData** que rellena la fuente.
DataSet, ResourceDataSet, DependencyDataSet	Las tablas de los elementos, los recursos y los vínculos. Solo la tabla de elementos es obligatoria.
ItemFields, ResourceFields, DependencyFields	Los nombres de las columnas. Los valores por defecto son los nombres de propiedad de los elementos (Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, RecurrenceRule, DependsOn), de los recursos (Id, Name, Color, Initials) y de los vínculos (FromId, ToId, Kind, Lag). Una columna que no existe no se asigna.
Active	Carga las tablas en los datos y las mantiene sincronizadas mientras sea true.
LiveUpdates	Las filas añadidas, modificadas o eliminadas en una tabla llegan a los datos de inmediato (true por defecto).
WriteBack	Las ediciones hechas en las vistas o en el código se escriben de vuelta en las filas (true por defecto).
OnLoadItem	Se dispara por cada elemento cargado, con la tabla y el `DataRow`, para rellenar campos que la asignación no cubre.

Ejemplo

Diferencias con TDataSet

- Un `DataTable` no tiene cursor. Un Edit y Post de Delphi es un `DataRow.BeginEdit` y `EndEdit`, o una asignación directa a la columna, y Cancel es `CancelEdit`. Una fila en edición se escribe de vuelta cuando termina su edición.
- La fuente escucha los eventos de la tabla: una fila añadida o modificada es un registro confirmado, una fila eliminada o una tabla vaciada quita los elementos, un cambio rechazado recarga el registro, y `AcceptChanges` se ignora. Las filas eliminadas y desasociadas se omiten.
- Las columnas escribibles son las que no son ReadOnly, AutoIncrement ni calculadas. Una columna de fecha numérica contiene fechas OLE Automation, y un texto vacío asignado a una columna que no es de texto establece `DBNull`.
- **OnLoadItem** pasa el `DataRow`, porque no hay registro actual. Los nombres de las columnas son **GetName** y **SetName** por índice, además de las propiedades con nombre.
- Vincule solo objetos `DataTable`: un `DataView` o un `IDataReader` no es una fuente. Una tabla liberada se desvincula sola, y la fuente es `IDisposable`.

Vincular los controles

Las propiedades de los controles son propiedades CLR simples, no propiedades de dependencia ni estilizadas, porque un plan es un modelo completo y no un único valor. En una aplicación MVVM mantenga el **TsgcPlannerData** en el modelo de vista y asígnelo a **Data** de los controles, y reaccione a **OnChange** o a un listener de los datos, a **SelectedId** y a los eventos de los controles.

Consulte también

- Descripción general de .NET
- Datos y programación en .NET
- Diferencias de .NET respecto a Delphi
- TsgcPlannerSource_DataSet
- Vinculación con DataSet

JSON, iCalendar y MS Project en .NET

Un plan se guarda y se carga en tres formatos, con los mismos resultados que en Delphi: los archivos que escribe una aplicación .NET los lee una aplicación Delphi y al revés.

Formato	Clases	Edición
JSON, el formato nativo con todos los campos	**SaveToJSON**, **LoadFromJSON**, **SaveToFile** y **LoadFromFile** de **TsgcPlannerData**, y las funciones de stream y de archivo de **sgcPlanner_JSON**.	Ambas
iCalendar (RFC 5545), para Outlook, Google Calendar y Apple Calendar	**TsgcPlannerICalendar**	Registered
Microsoft Project XML (MSPDI): tareas, vínculos, recursos, asignaciones y el calendario	**TsgcPlannerMSProject**	Registered

Ejemplo

Notas

- JSON se escribe como UTF-8 sin marca de orden de bytes; al cargar se acepta una marca. Un texto que no es UTF-8 válido dispara **EsgcPlannerError**.
- **TsgcPlannerICalendar** y **TsgcPlannerMSProject** son clases simples: créelas, establezca **Data** y llame a **ImportFromString**, **ImportFromStream** o **ImportFromFile**, y a **ExportToString**, **ExportToStream** o **ExportToFile**. **ImportMode** sustituye los datos o fusiona en ellos los elementos importados.
- Un problema en la entrada no detiene una importación: dispara **OnImportWarning** con la línea y un mensaje, y **WarningCount** los cuenta. Los mensajes son constantes públicas de las clases, con marcadores `{0}`.
- **ExportUTC** escribe las horas de iCalendar en UTC. **ProductId** es el PRODID de la exportación.
- MS Project XML se lee con `System.Xml.XmlReader`, con la DTD ignorada y sin resolvedor externo, y se escribe con el port del escritor de Delphi, de modo que la exportación es idéntica byte a byte a la de Delphi.

Consulte también

- Descripción general de .NET
- Datos y programación en .NET
- JSON
- iCalendar
- XML de MS Project

Renderizado sin ventana en .NET

Cada vista dibuja a través de su modelo de vista, y los modelos no necesitan ventana. Un servicio, una prueba o un informe puede cargar un plan y guardar el Scheduler, el Gantt y el tablero como PNG o SVG en cualquier plataforma, Linux incluido.

- **TsgcPlannerRenderer** en esegece.sgcPlanner.Rendering posee las tres vistas. Establezca **Data**, **View** (**prvScheduler**, **prvGantt**, **prvKanban**), **Width**, **Height**, **Scale**, **Theme** y las opciones de la vista, y luego llame a **SaveToPNG** o **SaveToSVG**.
- **SchedulerOptions**, **GanttOptions** y **KanbanOptions** tienen las propiedades de los controles. **Assign** las copia desde un modelo de vista, de modo que un informe puede dibujar lo que muestra un control. Lo que las opciones no cubren se establece en **SchedulerView**, **GanttView** y **KanbanView**.
- Un modelo de vista también puede dibujarse solo: **ExportImage** escribe SVG con esegece.sgcPlanner, y los métodos de extensión **SaveToPNG**, **ExportPNG** y **RenderToRaster** de **sgcPlanner_Raster** escriben PNG.

Ejemplo

Notas

- El tamaño está en píxeles lógicos; una escala de 2 escribe una imagen de 2400 x 1000 para una vista de 1200 x 500. **DeviceWidth** y **DeviceHeight** devuelven el tamaño de la imagen.
- **FontFile** y **FontFileBold** eligen un archivo de fuente TrueType, para un servidor sin la fuente por defecto.
- Las etiquetas de cabecera siguen el **TsgcPlannerLocale** compartido de `sgcPlanner_DateUtils.sgcPlannerLocale()`, tomado de la cultura actual en el primer uso. Para obtener la misma salida en todas las máquinas, establezca la cultura antes del primer renderizado, o establezca los nombres de ese locale.
- Los controles también tienen **SaveToPNG** y **SaveToSVG**, con el tamaño del control.

Consulte también

- Descripción general de .NET
- Clases de .NET
- TsgcPlannerRenderer
- Renderizado sin ventana

Sincronización con Google Calendar e iCalendar en .NET

Paquete: esegece.sgcPlanner.sgcWebSockets (edición Registered), espacio de nombres esegece.sgcPlanner. Se dirige a netstandard2.0, net8.0 y net9.0, y usa la biblioteca .NET de sgcWebSockets (esegece.sgcWebSockets) para HTTP y OAuth2 de Google.
El paquete .NET tiene los mismos dos componentes que el puente de Delphi, con los mismos nombres, la misma asignación, la misma regla de conflictos y el mismo formato de archivo de estado, de modo que un archivo de estado escrito por Delphi se carga en .NET y a la inversa.

- **TsgcPlannerGoogleSync** mantiene un TsgcPlannerData en sincronización bidireccional con un Google Calendar mediante un TsgcHTTP_Google_Cloud_Calendar_API_Client (su propiedad Client). Consulte TsgcPlannerGoogleSync.
- **TsgcPlannerICalSubscription** fusiona en los datos una URL iCalendar de solo lectura (https://, http:// o webcal://), analizada con TsgcPlannerICalendar. Consulte TsgcPlannerICalSubscription.

Diferencias con Delphi

- Ambas clases son clases simples que implementan IDisposable, no componentes. Dispose detiene el hilo de trabajo; después no se aplica nada.
- Los eventos son propiedades delegado, por ejemplo OnRefreshed, OnError, OnImportWarning y OnBeforeRefresh (un EventHandler). OnConflict y OnFilterItem reciben su resultado por ref.
- SyncAsync devuelve Task<bool>. RefreshAsync devuelve bool, False cuando hay una actualización en curso.
- El trabajo que toca Data y dispara los eventos se ejecuta en el SynchronizationContext del hilo que llamó a SyncAsync o RefreshAsync, o que asignó Active (el hilo de interfaz en WinForms, WPF y Avalonia). Sin contexto se ejecuta en el hilo de trabajo.
- LastSyncTime y LastRefreshTime son DateTime; default(DateTime) significa nunca.
- La costura de transporte de TsgcPlannerICalSubscription es el método virtual protegido DoFetch(string aURL, IList<string> aRequestHeaders, IList<string> aResponseHeaders, Stream aBody), que devuelve el estado HTTP. La implementación por defecto usa el HTTPClient propio (TsgcHTTP1Client) con HandleRedirects desactivado; el componente sigue por sí mismo hasta cinco redirecciones y envía If-None-Match y If-Modified-Since con los validadores de la última importación.
- No hay notificaciones de liberación: destruir Data no borra la propiedad Data.

Eventos de feed de solo lectura

Una actualización sobrescribe las ediciones locales de los elementos de la suscripción. Vételas en OnItemChanging, que las vistas consultan antes de confirmar un arrastre, un cambio de tamaño o una edición:

```
TsgcPlannerData oData = new TsgcPlannerData();
TsgcPlannerICalSubscription oSubscription = new TsgcPlannerICalSubscription();
oSubscription.Data = oData;
oSubscription.URL = "webcal://calendar.example.com/team.ics";
oSubscription.RefreshInterval = 60; // minutes
oSubscription.OnRefreshed = (s, aAdded, aUpdated, aDeleted) =>
  Console.WriteLine("added " + aAdded + ", updated " + aUpdated + ", deleted " + aDeleted);
oSubscription.OnError = (s, aError) => Console.WriteLine(aError);
oData.OnItemChanging = (object s, TsgcPlannerItem aItem, TsgcPlannerItemFields aFields, ref bool aAccept) =>
{
  if (oSubscription.IsSubscriptionItem(aItem))
    aAccept = false;
};
oSubscription.Active = true; // refreshes at once, then every hour
```

Sincronización con Google Calendar

```
TsgcHTTP_Google_Cloud_Calendar_API_Client oClient = new TsgcHTTP_Google_Cloud_Calendar_API_Client();
oClient.GoogleCloudOptions.OAuth2.ClientId = "your client id";
oClient.GoogleCloudOptions.OAuth2.ClientSecret = "your client secret";
oClient.GoogleCloudOptions.OAuth2.LocalPort = 8080;
oClient.GoogleCloudOptions.OAuth2.Scope.Add("https://www.googleapis.com/auth/calendar");

TsgcPlannerGoogleSync oSync = new TsgcPlannerGoogleSync();
oSync.Client = oClient;
oSync.Data = oData;
oSync.CalendarId = "primary";
oSync.StateFileName = "calendar_state.json";
oSync.OnSyncDone = (s, aPulled, aPushed, aDeleted, aConflicts) =>
  Console.WriteLine("pulled " + aPulled + ", pushed " + aPushed);
oSync.Active = true;
bool vOK = await oSync.SyncAsync();
```

Los elementos cuyo Id empieza por ExcludeIdPrefix (ics: por defecto, el IdPrefix de TsgcPlannerICalSubscription) nunca se envían a Google.

Demo

La demo de WinForms demos\Planner\GoogleCalendarSync muestra ambos componentes en un TsgcWinFormsScheduler. Ejecútela con --test para una autoprueba sin conexión: un transporte de Google de prueba y un servidor iCalendar de bucle local, sin necesidad de credenciales.

Véase también

- TsgcPlannerGoogleSync
- TsgcPlannerICalSubscription
- Sincronización con Google Calendar
- TsgcPlannerICalendar
- TsgcPlannerData
- iCalendar

Demos de .NET

Los demos están en `.net\demos\Planner`, una carpeta por demo con su proyecto, su `Program.cs` y un README.md. Construyen su interfaz de usuario en código, sin XAML. Cada demo tiene un conmutador `--test` que ejecuta una autocomprobación fuera de pantalla, escribe líneas PASS y FAIL y termina con el número de comprobaciones fallidas, como el conmutador `/test` de los demos de Delphi, y los demos con ventana tienen un conmutador `--snapshot` que renderiza las vistas a archivos PNG.

Demo	Host	Muestra
ProjectPlanner	WinForms	Un proyecto de relanzamiento de un sitio web mostrado de tres maneras a la vez: un diagrama de Gantt, un tablero Kanban y una semana, todos vinculados a los mismos datos. Catorce tareas en dos pistas, cinco personas, cuatro estados y vínculos de fin a inicio. Arrastre una tarjeta en el tablero y el Gantt y el Scheduler la siguen.
ProjectPlannerWpf	WPF	La misma ventana con los controles **TsgcWpf**.
ProjectPlannerAvalonia	Avalonia	La misma ventana con los controles **TsgcAvalonia**, para Windows, Linux y macOS.
TeamScheduler	WinForms	Tres semanas de reuniones de equipo para cinco personas: día, semana laboral, semana, mes, agenda y la Timeline de recursos, una reunión diaria de pie y una sincronización semanal como series recurrentes, una conferencia de día completo, arrastrar para crear con **OnItemCreate** y agrupación por recurso.
TeamSchedulerWpf	WPF	La misma ventana con **TsgcWpfScheduler**.
TeamScheduleAvalonia	Avalonia	La misma ventana con **TsgcAvaloniaScheduler**.
HeadlessRender	Consola	Sin ventana: carga un proyecto desde JSON (o construye uno y lo guarda como project.json) y escribe un PNG y un SVG del Scheduler, del Gantt y del tablero con **TsgcPlannerRenderer**, en Windows, Linux y macOS. El equivalente .NET del demo HeadlessReport de Delphi.
GoogleCalendarSync	WinForms	Una vista semanal mantenida en sincronización bidireccional con un calendario de Google, más una suscripción iCalendar de solo lectura, con los componentes del puente esegece.sgcPlanner.sgcWebSockets (Registered).

La parte sin interfaz de ProjectPlanner y TeamScheduler (los datos generados, la configuración de las vistas, el snapshot y la autocomprobación) está en `ProjectPlanner_Shared.cs` y `TeamScheduler_Shared.cs`, que compilan los proyectos WinForms, WPF y Avalonia, de modo que los tres hosts ejecutan las mismas comprobaciones.
Cada demo se compila y se ejecuta en las dos ediciones. En la edición Community TeamScheduler no tiene la entrada Timeline y muestra cada serie una sola vez, ProjectPlanner no dibuja flechas de dependencia, y GoogleCalendarSync explica que el puente es un paquete Registered.

Compilar los demos

Dentro del árbol del producto los demos se compilan contra los proyectos de la biblioteca. El zip de los demos usa por defecto los paquetes NuGet de la edición Community. Compile un demo con los paquetes Registered estableciendo **SgcPlannerEdition**:

```
dotnet build ProjectPlanner\ProjectPlanner.csproj -p:SgcPlannerEdition=Registered
dotnet run --project HeadlessRender\HeadlessRender.csproj -- --test
```

Consulte también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Demos

Solución de problemas

EsgcPlannerError: This feature requires the Registered edition of sgcPlanner

Schedule y SetBaseline del Gantt lo provocan en un build Community. Ponga esas llamadas dentro de `{$IFDEF SGC_PLANNER_ADVANCED}`, o use la edición Registered o Trial. Consulte Características por edición.

Un formulario no carga en un build Community: Class TsgcPlannerICalendar not found

TsgcPlannerICalendar y TsgcPlannerMSProject no forman parte de la edición Community. Créelos en el código dentro de `{$IFDEF SGC_PLANNER_ADVANCED}` en lugar de soltarlos en un formulario que también deba cargar en Community.

El Scheduler no muestra eventos

- Compruebe que los eventos caen en el periodo en pantalla. Date 0 sigue el reloj, de modo que la vista muestra la semana actual; defina Date o llame a GoToday, Next y Prior.
- Las vistas de día muestran las horas de Options.DayStartHour a Options.DayEndHour. Un evento fuera de ellas se dibuja recortado en el borde.
- Un evento cuyo Finish es anterior a su Start se lee como un evento de duración cero en su Start.
- Asegúrese de que la propiedad Data del Scheduler es el objeto de datos que rellena su código. Un control sin Data usa un objeto de datos interno propio.

Un evento recurrente aparece solo una vez

La recurrencia forma parte de la edición Registered. En Registered, compruebe la regla con TsgcPlannerRecurrence.TryParse, que devuelve el motivo cuando una regla no es compatible, y compruebe COUNT, UNTIL y RecurrenceExceptions. Consulte Recurrencia.

La programación automática no mueve una tarea

- El motor conserva las fechas de las tareas Manual, los elementos recurrentes y las tareas de un ciclo de dependencias. OnScheduleError enumera las tareas de un ciclo y Schedule devuelve False.
- Una tarea con Constraint pcSNET no comienza antes de su ConstraintDate. Un movimiento en el Gantt establece esta restricción; vuelva a poner Constraint en pcASAP para que el motor adelante la tarea.
- AutoSchedule solo se ejecuta tras una edición hecha en el Gantt. Tras un cambio en el código o en otra vista, llame a Schedule.

Consulte Programación automática.

El Gantt no dibuja flechas de dependencia

Las flechas necesitan la edición Registered y ShowDependencies True. Un vínculo solo se dibuja cuando sus dos tareas están en filas visibles, de modo que un vínculo hacia un resumen plegado queda oculto, y un vínculo con FromId o ToId desconocido nunca se dibuja.

Un arrastre no hace nada o el elemento vuelve atrás

Compruebe ReadOnly y DragOptions del control, y los manejadores que pueden rechazar una edición: OnItemChanging de los datos, OnCardMoving, OnItemCreate, OnLinkCreating y OnRecurrenceEdit. En el tablero, una columna en su límite de WIP con WIPMode pwmBlock rechaza tarjetas nuevas, y la columna desconocida nunca es un destino de drop. En el Scheduler, los elementos resumen se muestran pero no se pueden arrastrar. Consulte Arrastrar y soltar.

EsgcPlannerError: OnChange kept changing the planner data for more than 8 rounds

Un manejador de OnChange, o un listener, cambia los datos en cada notificación, de modo que los cambios nunca se estabilizan. Cambie los datos solo cuando difieran del valor que desea, o saque el cambio del manejador.

Las horas están desfasadas unas horas tras una importación o exportación iCalendar

Las horas en UTC o con zona horaria se convierten a la hora local de la máquina al importar. Al exportar, ExportUTC escribe horas UTC para los elementos que no se repiten, que un calendario muestra en su propia zona horaria. Ponga ExportUTC en False para escribir horas locales. Consulte iCalendar.

Las fechas, duraciones o nombres se ven distintos en otra máquina o en Linux

Las etiquetas y los números de las duraciones usan sgcPlannerLocale, que se inicializa desde la máquina. Defina sus propiedades, por ejemplo ShortDateFormat, Use24Hour y DecimalSeparator, al inicio para obtener la misma salida en todas partes.

Se pierden caracteres en Delphi 7 o Delphi 2007

Estas versiones tienen cadenas Ansi, de modo que el texto UTF-8 de los archivos JSON, iCalendar y XML de MS Project se convierte a la página de códigos del sistema, y los caracteres fuera de ella se pierden. Delphi 2009 y posteriores conservan todos los caracteres.

Las ediciones no se escriben de vuelta en el dataset

- Compruebe que WriteBack es True y que IdField nombra un campo que Locate pueda encontrar.
- No se escribe nada mientras el dataset está en estado de edición o inserción; los cambios se escriben después de Post o Cancel.
- Los recursos, los campos calculados, los campos de lookup y los campos de solo lectura nunca se escriben.

Consulte Vinculación con DataSet.

LoadFromJSON provoca EsgcPlannerError

El texto no es JSON válido, no es un documento sgcPlanner, o se escribió con una versión más nueva del formato. Los datos quedan como estaban, de modo que la aplicación puede notificar el mensaje y seguir ejecutándose. Consulte Persistencia JSON.

Una violación de acceso después de liberar un formulario o un objeto

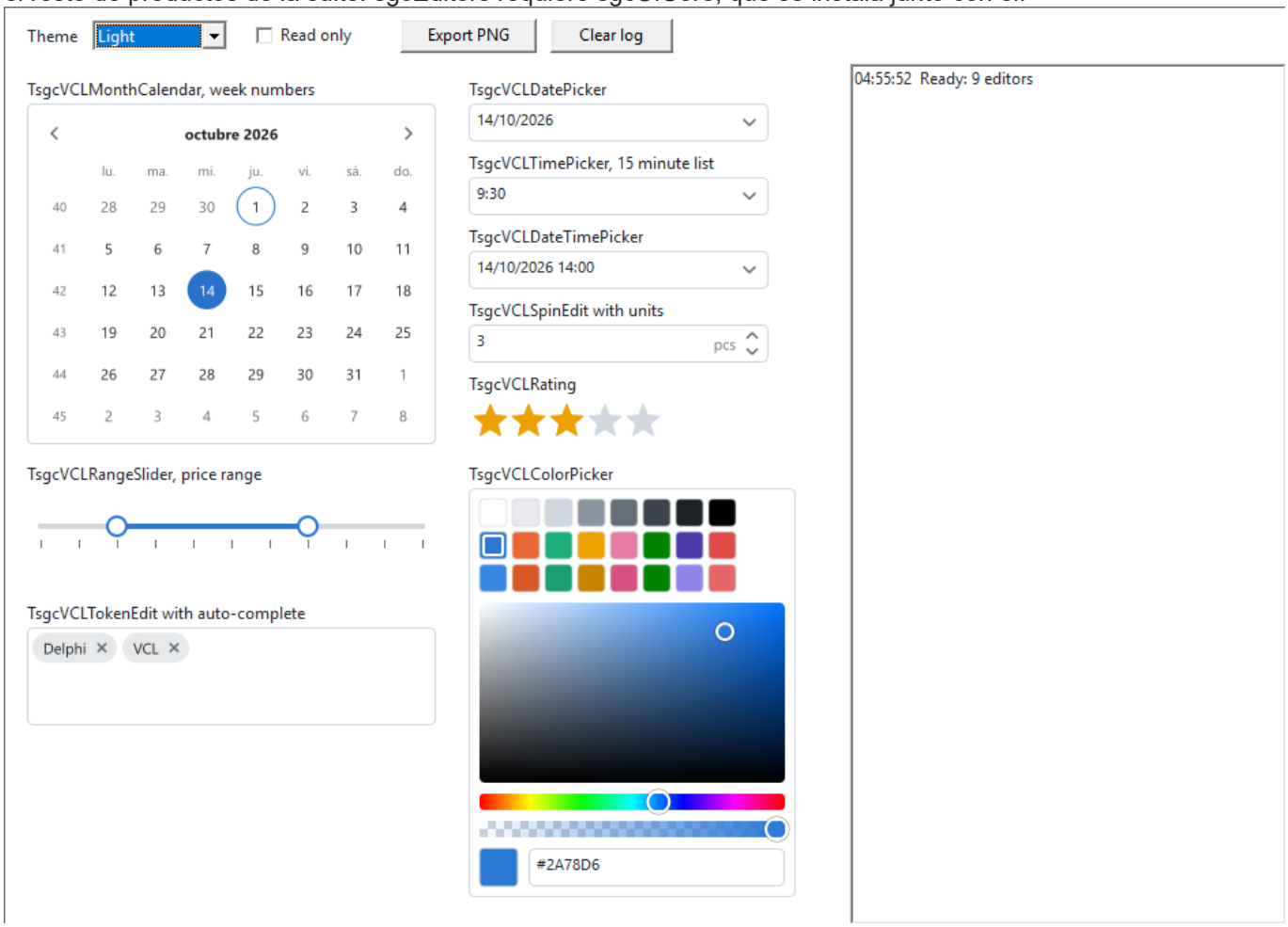
Un manejador añadido con AddListener debe quitarse con RemoveListener antes de liberar el objeto que lo posee. Los datos no saben cuándo desaparece ese objeto.

Consulte también

- Características por edición
- Plataformas
- TsgcPlannerData

Introducción

sgcEditors añade a las aplicaciones Delphi y C++Builder los controles de entrada de un formulario, para VCL y FireMonkey: un calendario mensual, un selector de fecha, un selector de hora, un selector de fecha y hora, un control deslizante de rango, un selector de color, un token edit, una valoración y un spin edit. Cada control existe para VCL, con el prefijo TsgcVCL, y para FireMonkey, con el prefijo TsgcFMX. Cinco variantes data-aware enlazan el selector de fecha, el selector de hora, el spin edit, la valoración y el control deslizante de rango con un campo de un TDataSet.
El texto se escribe siempre en un edit nativo, nunca en un cursor propio, de modo que la composición de IME, el portapapeles, deshacer y los lectores de pantalla funcionan como en cualquier otro edit del formulario. Todo lo demás, el marco, los botones, el calendario, los chips, las estrellas, los tiradores y el cuadrado de color, lo dibuja **sgcUICore**, así que un control tiene el mismo aspecto en VCL, en FireMonkey y en la salida PNG y SVG sin ventana. Los controles tienen temas claro y oscuro, un color de acento, soporte completo de teclado, un nombre accesible y escalado de DPI por monitor.
Detrás de cada control hay una clase de modelo independiente del framework, como TsgcCalendarModel o TsgcSpinModel. Los controles VCL y FireMonkey solo traducen la entrada y alojan el edit y la lista desplegable, de modo que el mismo modelo también se renderiza sin ventana, en un servidor o en Linux64.
sgcEditors forma parte de la suite visual de eSeGeCe. Se dibuja con sgcUICore, el paquete de runtime compartido con el canvas, el rasterizador, las fuentes, el escritor de SVG y los temas, de modo que los editores combinan con el resto de productos de la suite. sgcEditors requiere sgcUICore, que se instala junto con él.



Ediciones

sgcEditors se distribuye como Registered, Trial y Community. Community conserva los nueve controles para VCL y FireMonkey y deja fuera los cinco controles data-aware, el cuadrado HSV y la barra alfa del selector de color, la lista de autocompletado del token edit y el segundo tirador del control deslizante de rango. Consulte Ediciones.

Por dónde empezar

- Arquitectura: los modelos, los hosts, el edit interno, la lista desplegable y las dos capas de renderizado.
- Controles: los nueve controles, sus variantes data-aware y sus clases de modelo de un vistazo.
- Las páginas de los controles: Calendario mensual, Selector de fecha, Selector de hora, Selector de fecha y hora, Control deslizante de rango, Selector de color, Token edit, Valoración y Spin edit, los controles data-aware y las clases de modelo, con sus propiedades, métodos, eventos y ejemplos.
- Instalación y Plataformas: el instalador, los paquetes, las versiones de Delphi y C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX, Inicio rápido C++Builder e Inicio rápido data-aware: un primer formulario en pocos minutos.
- Entrada de texto e IME, Teclado y accesibilidad, Temas, Localización, DPI, Salida sin ventana y Ediciones.
- Solución de problemas: problemas habituales y sus causas.

Arquitectura

sgcEditors mantiene la lógica de edición separada del framework. Cada familia de controles tiene una clase de modelo que solo usa la RTL y sgcUICore, y los controles VCL y FireMonkey son hosts delgados a su alrededor.

Las partes de un control

1. **Modelo.** Un descendiente TPersistent de TsgcEditorModel, por ejemplo TsgcCalendarModel, TsgcDatePickerModel o TsgcTokenModel. Guarda el valor y las opciones, recibe las teclas y el ratón en píxeles lógicos mediante KeyDown, KeyPress, MouseDown, MouseMove, MouseUp y MouseWheel, y se dibuja en cualquier canvas de sgcUICore. Funciona sin ventana, de modo que también es el renderizador sin ventana.
2. **Host.** El control VCL desciende de TsgcVCLCustomEditor y el control FireMonkey de TsgcFMXCustomEditor. El host crea el modelo, le envía la entrada, pinta sus dos capas, dispara OnChange y publica las propiedades del modelo. El modelo de un control es public como Model, y cada control tiene además una propiedad tipada para él, como CalendarModel, PickerModel o SpinModel.
3. **Edit interno.** Los selectores, el spin edit, el selector de color y el token edit alojan un edit nativo sin borde: un TEdit hijo en VCL y un TEdit de FMX en FireMonkey. El modelo indica dónde va el texto, y el host coloca allí el edit. Consulte Entrada de texto e IME.
4. **Lista desplegable.** El calendario del selector de fecha, la lista de horas del selector de hora, ambos lado a lado para el selector de fecha y hora, y la lista de autocompletado del token edit se abren en una lista desplegable por framework. En VCL es una ventana de herramientas WS_POPUP que nunca quita el foco al edit, se abre debajo del control, pasa arriba cuando no hay sitio y se queda dentro del área de trabajo del monitor. En FireMonkey es un TPopup colocado debajo del control.

Dos capas

Un modelo dibuja dos capas. Render dibuja la capa de contenido: el marco, el texto, los días, las muestras, los chips y las marcas. RenderOverlay dibuja encima lo que cambia a menudo: el hover, el anillo de foco, los tiradores del control deslizante, el cursor del cuadrado de color, el relleno de la valoración y su vista previa de hover. Un arrastre, un hover o un cambio de foco vuelve a dibujar solo el overlay, de modo que la capa de contenido no se rasteriza de nuevo. Cada cambio de un modelo se comunica mediante OnModelChange con uno de estos tipos:

Tipo	El host
eckContent	Renderiza de nuevo la capa de contenido.
eckOverlay	Renderiza solo el overlay.
eckValue	Dispara OnChange y, en VCL, notifica el nuevo valor a los lectores de pantalla.
eckLayout	Mueve el edit interno, escribe de nuevo su texto y renderiza la capa de contenido.

Units

Unidad	Contenido
sgcEditors_Types	Las enumeraciones, TsgcEditorModel, TsgcEditorsPainter (colores del tema, marco, anillo de foco, chevrons, flechas de spin), códigos de tecla, formato e interpretación de números, conversión HSV y hexadecimal.
sgcEditors_Date	Números de día, números de semana ISO 8601 y simples, la cuadrícula del mes, TsgcEditorLocale y el formato, el paso y la interpretación de fechas.

sgcEditors_Calendar, _Picker, _Slider, _Spin, _Color, _Rating, _Token, _List	Los modelos. sgcEditors_Picker contiene además TsgcDateTimeDropDownModel, el calendario y la lista de horas de la lista desplegable de fecha y hora.
sgcEditors_VCL, _VCL_Date, _VCL_Value, _VCL_Color, _VCL_Token	La base del host VCL, el edit interno, la lista desplegable y los controles VCL. Solo Windows.
sgcEditors_FMX, _FMX_Date, _FMX_Value, _FMX_Color, _FMX_Token	Lo mismo para FireMonkey, desde Delphi XE6.
sgcEditors_Data, _VCL_DB, _FMX_DB	TsgcEditorFieldLink y los controles data-aware. Edición Advanced, estas unidades están vacías en Community.
sgcEditors_Const, _License, _Reg	Las constantes de versión y edición, la comprobación de prueba y el registro en tiempo de diseño.

Hilos

Los controles se usan desde el hilo principal, como cualquier otro control VCL o FireMonkey, y sus eventos se lanzan en él.

Rendimiento

El grupo de rendimiento de QA mide estos escenarios con una compilación release de Delphi 13 en Win64. El presupuesto es el límite de cada escenario, y la columna medida es una ejecución en la máquina de referencia. El calendario mensual tiene además un presupuesto Win32 de 2,500 µs, y Linux64 informa E1, E4 y E8 sin límite.

ID	Escenario	Presupuesto	Medido
E1	Contenido del calendario mensual, 280 x 240: título, nombres de los días, 42 días, números de semana y una banda de rango	1,500 µs	395 µs
E2	Solo overlay del calendario: celda en hover y anillo de foco	60 µs	24 µs
E3	IsDateEnabled sobre 10,000 días con 500 fechas deshabilitadas, 2 días de la semana deshabilitados y el evento asignado	2 ms	0.3 ms
E4	Control deslizante de rango 300 x 48 con 11 marcas y etiquetas, renderizado completo; un arrastre de tirador, solo overlay	400 µs; 150 µs	88 µs; 73 µs
E5	Selector de color 240 x 300 con 24 muestras, el cuadrado, las barras de tono y alfa, renderizado completo; un arrastre en el cuadrado, solo overlay	1,500 µs; 150 µs	1,018 µs; 114 µs
E6	Token edit con 50 chips, 400 x 120, distribución y renderizado; el filtro de autocompletado de 10,000 elementos por pulsación	1,500 µs; 3 ms	436 µs; 0.2 ms
E7	Valoración, 5 estrellas 160 x 32, renderizado completo; marco del spin edit 120 x 28	200 µs; 120 µs	103 µs; 39 µs
E8	100,000 formatos de fecha más 100,000 interpretaciones; 1,000,000 de números de semana ISO	150 ms; 100 ms	23 ms; 27 ms
E9	Un formulario VCL con 30 de cada uno de los 9 controles, creado y pintado por primera vez; una lista desplegable de selector abierta hasta su primer fotograma	350 ms; 16 ms	unos 340 ms en una máquina

			tranquila; 5 ms
E10	El formulario E9 en FireMonkey	solo informado	

La mayor parte del tiempo de E9 es la creación de las ventanas de los 270 controles y de sus edits internos, no el dibujo. Un formulario que muestra cientos de editores a la vez se abre más rápido cuando los controles de las páginas ocultas se crean bajo demanda.

Véase también

- Controles
- TsgcEditorModel
- Salida sin ventana
- Entrada de texto e IME

Controles

sgcEditors tiene nueve controles. Cada uno existe para VCL y para FireMonkey con las mismas propiedades y eventos publicados.

Control	VCL	FireMonkey	Entrada de texto
Calendario mensual	TsgcVCLMonthCalendar	TsgcFMXMonthCalendar	Dibujado. Vistas de mes, año y década, fecha única o rango, límites, fechas y días de la semana deshabilitados, números de semana.
Selector de fecha	TsgcVCLDatePicker	TsgcFMXDatePicker	Edit nativo más un calendario desplegable.
Selector de hora	TsgcVCLTimePicker	TsgcFMXTimePicker	Edit nativo más una lista desplegable de horas cada ListInterval minutos.
Selector de fecha y hora	TsgcVCLDateTimePicker	TsgcFMXDateTimePicker	Edit nativo más el calendario y la lista de horas lado a lado.
Control deslizante de rango	TsgcVCLRangeSlider	TsgcFMXRangeSlider	Dibujado. Uno o dos tiradores, pasos, marcas, etiquetas y una sugerencia de valor, horizontal o vertical.
Selector de color	TsgcVCLColorPicker	TsgcFMXColorPicker	Muestras, cuadrado HSV y barras de tono y alfa dibujados, más un edit hexadecimal nativo.
Token edit	TsgcVCLTokenEdit	TsgcFMXTokenEdit	Chips dibujados, más un edit nativo para el siguiente token y una lista de autocompletado.
Valoración	TsgcVCLRating	TsgcFMXRating	Dibujado. Estrellas, corazones, círculos o un glifo propio, valores enteros, medios o exactos.
Spin edit	TsgcVCLSpinEdit	TsgcFMXSpinEdit	Edit nativo más botones de subir y bajar, unidades, límites y repetición automática.

Controles data-aware

Los controles data-aware descienden del control de la segunda columna y añaden DataSource, DataField y un Field público. Forman parte de las ediciones Registered y Trial. Consulte Inicio rápido data-aware.

VCL	FireMonkey	Control base	Enlaza
TsgcVCLDBDatePicker	TsgcFMXDBDatePicker	Selector de fecha	La fecha del campo, se conserva su parte de hora.
TsgcVCLDBTimePicker	TsgcFMXDBTimePicker	Selector de hora	La hora del campo, se conserva su parte de fecha.
TsgcVCLDBSpinEdit	TsgcFMXDBSpinEdit	Spin edit	Un campo numérico. Un campo entero cambia el spin edit a esvInteger.
TsgcVCLDBRating	TsgcFMXDBRating	Valoración	Un campo numérico.

TsgcVCLDBRangeSlider	TsgcFMXDBRangeSlider	Control deslizante de rango	Value con DataField y, en modo de rango, EndValue con EndDataField.

Lo que tiene cada control

Publicados: Theme, AccentColor, TextStyle, ReadOnly, AccessibleName y OnChange, además de las propiedades y eventos habituales del framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Public: Model, FormatSettings, DroppedDown, DropDown, CloseUp, SaveToPNG y SaveToSVG.

Clases de modelo

Todos los modelos descienden de TsgcEditorModel.

Modelo	Usado por
TsgcCalendarModel	El calendario mensual y la lista desplegable de los selectores de fecha.
TsgcDatePickerModel	Los selectores de fecha, de hora y de fecha y hora. Kind es edpDate, edpTime o edpDateTime.
TsgcDateTimeDropDownModel	La lista desplegable del selector de fecha y hora: un calendario y una lista de horas.
TsgcEditorListModel	La lista de horas y la lista de autocompletado.
TsgcSliderModel	El control deslizante de rango.
TsgcColorPickerModel	El selector de color.
TsgcTokenModel	El token edit.
TsgcRatingModel	La valoración.
TsgcSpinModel	El spin edit.

Véase también

- Arquitectura
- Inicio rápido VCL

Instalación

sgcEditors forma parte de la distribución de sgcUI. Un instalador cubre los productos de sgcUI e instala solo los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcEditors** y los demás productos que desee. El servidor de licencias devuelve los productos que posee, y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos, después compila los paquetes en su máquina para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué incluye sgcEditors

sgcUICore, la base compartida de los productos visuales de sgcUI, se incluye con sgcEditors. No tiene casilla propia. El instalador lo instala automáticamente siempre que se selecciona sgcEditors.

Carpetas instaladas

Cada producto se instala en una sola carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de todos los productos instalados. Esta es la única ruta de biblioteca de sgcUI.
`Packages`	Los paquetes de todos los productos instalados, un conjunto por producto y por versión del IDE.
`Lib<V>`	Las unidades compiladas para cada versión del IDE y plataforma, por ejemplo `LibD13` .
`Demos\Editors`	Las demos de sgcEditors: Gallery, GalleryFMX, Calendar, BookingForm y DataAware.

El instalador añade `sgcUI\Source` a la ruta de biblioteca y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las unidades compiladas. sgcUICore vive en la misma carpeta `Source`, de modo que no necesita una ruta propia.

La paleta de componentes

Los componentes están en la página de paleta **SGC Editors**. Un formulario VCL muestra los controles VCL y un formulario FireMonkey los controles FMX. Las ediciones Registered y Trial registran 28 componentes, los nueve controles y los cinco controles data-aware de cada framework. Community registra los 18 controles sin los data-aware.

Trial y Community

- **Trial**: el instalador trial de sgcUI instala todos los productos en modo de prueba, sgcEditors incluido.

- **Community**: el instalador gratuito sgcUI Community incluye sgcEditors sin sus características avanzadas. Consulte Ediciones.

Compilar los paquetes manualmente

Para compilar los paquetes usted mismo, ábralos para su versión del IDE desde `sgcUI\Packages`. `<V>` es el sufijo de la versión de Delphi, por ejemplo `sgcEditorsD13.dpk`.

Paquete	Contenido
`sgcUICore<V>`	La base compartida sgcUICore (runtime).
`sgcEditors<V>` / `dclsgcEditors<V>`	Los componentes de sgcEditors (runtime / tiempo de diseño).

Compílelos en orden de dependencia: `sgcUICore`, después `sgcEditors`, después `dclsgcEditors`. A continuación instale el paquete de tiempo de diseño `dclsgcEditors` y añada `sgcUI\Source` a la ruta de biblioteca de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra el proyecto `.cbproj` que corresponda a su versión, por ejemplo `sgcEditorsC13.cbproj`, y compílelo. Los encabezados `.hpp` generados aparecen junto al paquete compilado.

Véase también

- Plataformas
- Inicio rápido VCL

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13. Los paquetes generan los encabezados .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza.
VCL	Los controles TsgcVCL en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	Los controles TsgcFMX desde Delphi XE6 (SGC_EDITORS_FMX), en las plataformas FireMonkey. Solo usan FireMonkey y sgcUICore.
Linux64	Las unidades independientes del framework desde Delphi 10.2 (SGC_EDITORS_LINUX): los modelos, las matemáticas de fechas y la configuración regional, con la salida PNG y SVG sin ventana. No hay controles para Linux.
.NET	Una versión en C# con la misma API para WinForms, WPF y Avalonia, los nueve controles en todos los hosts y la salida PNG y SVG sin ventana, en .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Paquetes

sgcEditors incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcEditorsD13.dpk para Delphi 13 y sgcEditorsC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcEditors	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey.	rtl, dbrtl, sgcUICore
sgcEditorsVCL	Runtime, VCL	sgcEditors_VCL, sgcEditors_VCL_Color, sgcEditors_VCL_DB, sgcEditors_VCL_Date, sgcEditors_VCL_Token y sgcEditors_VCL_Value	rtl, vcl, dbrtl, sgcUICore, sgcUICoreVCL, sgcEditors
sgcEditorsFMX	Runtime, FireMonkey, desde XE6	sgcEditors_FMX, sgcEditors_FMX_Color, sgcEditors_FMX_DB, sgcEditors_FMX_Date, sgcEditors_FMX_Token y sgcEditors_FMX_Value	rtl, fmx, dbrtl, sgcUICore, sgcUICoreFMX, sgcEditors
dclsgcEditors	Tiempo de diseño, VCL	Registra los editores VCL en la página de paleta **SGC Editors**, con el editor estándar de listas de cadenas para DisabledDates y Swatches.	designide, rtl, vcl, sgcEditors, sgcEditorsVCL
dclsgcEditorsFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra los editores FireMonkey en la página de paleta **SGC Editors**.	designide, rtl, fmx, sgcEditors, sgcEditorsFMX, dclsgcEditors

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcEditorsFMX requiere dclsgcEditors, así que instale primero dclsgcEditors.

Versiones del IDE y plataformas de destino

Delphi	sgcEditors	sgcEditorsVCL	sgcEditorsFMX	dclsgcEditors, dclsgcEditorsFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcEditors, sgcEditorsVCL	sgcEditorsFMX	dclsgcEditors, dclsgcEditorsFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcEditors)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcEditors)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcEditorsC13.cbproj, sgcEditorsVCLC13.cbproj, sgcEditorsFMXC13.cbproj, dclsgcEditorsC13.cbproj y dclsgcEditorsFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (los editores TsgcFMX) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcEditorsVCL solo funcionan en Windows.

- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcEditors_FMX, sgcEditors_FMX_Color, sgcEditors_FMX_DB, sgcEditors_FMX_Date, sgcEditors_FMX_Token y sgcEditors_FMX_Value.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcEditors se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa los modelos de los editores (valores, fechas, colores, tokens). No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Cuadro de texto interno de los editores VCL

Los editores VCL con un campo de texto crean su cuadro de texto interno (una ventana de edición nativa) bajo demanda, lo que hace que los formularios con muchos editores se muestren rápido. Hasta entonces el editor dibuja el texto él mismo. El cuadro de texto interno se crea al pasar el ratón por primera vez, cuando el editor recibe el foco, en el primer uso del teclado o del portapapeles, o cuando el código lee su handle de ventana. Las herramientas de accesibilidad, como los lectores de pantalla y los clientes de UI Automation, ven el cuadro de texto interno solo a partir de ese momento. El propio editor responde a WM_GETTEXT desde el principio. Un editor deshabilitado o ReadOnly, o con PasswordChar, lectura de derecha a izquierda, TextHint o una Alignment distinta de izquierda, lo crea de inmediato, y en tiempo de diseño siempre se crea.

Restricciones de versión

- Las unidades independientes del framework y las unidades VCL compilan en Delphi 7: sin genéricos, sin métodos anónimos, sin variables inline y sin record helpers.
- TextHint de los controles VCL se muestra desde Delphi 2009.
- Las enumeraciones tienen prefijos de editor (esm, efd, ewn, ecv, eor, etp, ecp, eti, emm, erp, erg, esv, esb), de modo que los encabezados de C++Builder nunca chocan con los demás productos de sgcUI.

Véase también

- Ediciones
- Arquitectura

Inicio rápido VCL

Instale los paquetes de sgcEditors para su versión de Delphi. Los componentes están en la página de paleta **SGC Editors**.

En tiempo de diseño

1. Coloque un **TsgcVCLDatePicker** en un formulario. Asigne MinDate y Date en el Object Inspector, o deje Date vacío con AllowEmpty en True.
2. Coloque un **TsgcVCLSpinEdit**, asigne MinValue, MaxValue y Value, y escriba un sufijo como guests en Units.
3. Coloque un **TsgcVCLTokenEdit** y asígnele un TextHint.
4. Asigne un único manejador OnChange a los tres controles y ejecute la aplicación. Escriba una fecha o abra el calendario con F4 o Alt+Down, cambie el spin edit con las flechas o la rueda del ratón, y escriba tokens separados por comas.

En código

Un control funciona en cuanto tiene un Parent. Asigne los límites de un spin edit antes de su Value, para que el valor no quede acotado a los límites antiguos.

```
uses
  SysUtils, Classes, Controls, Forms, StdCtrls,
  sgcEditors_VCL_Date, sgcEditors_VCL_Value, sgcEditors_VCL_Token;

type
  TBooking = class(TComponent)
  public
    Arrival: TsgcVCLDatePicker;
    Guests: TsgcVCLSpinEdit;
    Tags: TsgcVCLTokenEdit;
    Summary: TLabel;
    procedure EditorChange(Sender: TObject);
  end;

procedure TBooking.EditorChange(Sender: TObject);
begin
  // OnChange fires for every new value, typed, picked or stepped
  if Arrival.IsEmpty then
    Summary.Caption := 'No arrival date'
  else
    Summary.Caption := Format('%s, %d guests, %d tags',
      [DateToStr(Arrival.Date), Round(Guests.Value), Tags.Tokens.Count]);
end;

function CreateBooking(aForm: TForm): TBooking;
begin
  Result := TBooking.Create(aForm);

  Result.Arrival := TsgcVCLDatePicker.Create(aForm);
  Result.Arrival.Parent := aForm;
  Result.Arrival.SetBounds(16, 16, 160, 28);
  Result.Arrival.MinDate := Date;
  Result.Arrival.Date := Date + 1;
  Result.Arrival.OnChange := Result.EditorChange;

  // the limits before the value, MinValue = MaxValue means no limit
  Result.Guests := TsgcVCLSpinEdit.Create(aForm);
  Result.Guests.Parent := aForm;
  Result.Guests.SetBounds(16, 56, 120, 28);
```

```
  Result.Guests.MinValue := 1;
  Result.Guests.MaxValue := 12;
  Result.Guests.Value := 2;
  Result.Guests.Units := 'guests';
  Result.Guests.OnChange := Result.EditorChange;

  Result.Tags := TsgcVCLTokenEdit.Create(aForm);
  Result.Tags.Parent := aForm;
  Result.Tags.SetBounds(16, 96, 320, 64);
  Result.Tags.TextHint := 'Add a tag';
  Result.Tags.AddToken('Late arrival');
  Result.Tags.OnChange := Result.EditorChange;

  Result.Summary := TLabel.Create(aForm);
  Result.Summary.Parent := aForm;
  Result.Summary.SetBounds(16, 172, 320, 20);
  Result.EditorChange(nil);
end;
```

Véase también

- Controles
- Entrada de texto e IME
- Teclado y accesibilidad

Inicio rápido FMX

Los controles FireMonkey necesitan Delphi XE6 o posterior. Están en la página de paleta **SGC Editors** de un formulario FireMonkey y tienen las mismas propiedades y eventos que los controles VCL.

En tiempo de diseño

1. Coloque un **TsgcFMXDatePicker**, un **TsgcFMXRangeSlider** y un **TsgcFMXColorPicker** en un formulario.
2. En el control deslizante, asigne True a RangeMode, después Min, Max, Step y TickInterval, y active ShowLabels.
3. En el selector de color, elija las Parts que mostrar y el SelectedColor.

Qué es distinto en FireMonkey

- La lista desplegable es un TPopup debajo del control. Cuando se cierra con un clic fuera de ella, se deshacen las elecciones hechas mientras estaba abierta. Enter, o la elección que completa el valor, la cierra y conserva el valor.
- FireMonkey mueve el foco con Tab antes de que un control vea la tecla, de modo que el control deslizante de rango y el selector de color usan Ctrl con las flechas para moverse entre sus partes. Consulte Teclado y accesibilidad.
- El texto accesible de un control va a su Hint, respetando cualquier Hint propio. TabStop, Hint y ShowHint se publican desde Delphi 10.1.
- El edit interno del spin edit pide el teclado virtual NumbersAndPunctuation en las plataformas móviles.

En código

```
uses
  SysUtils, Classes, FMX.Types, FMX.Controls, FMX.Forms, FMX.StdCtrls,
  sgcEditors_Types, sgcEditors_FMX_Date, sgcEditors_FMX_Value,
  sgcEditors_FMX_Color;

type
  TFilters = class(TComponent)
  public
    Day: TsgcFMXDatePicker;
    Price: TsgcFMXRangeSlider;
    Colour: TsgcFMXColorPicker;
    Info: TLabel;
    procedure PriceComplete(Sender: TObject);
  end;

procedure TFilters.PriceComplete(Sender: TObject);
begin
  // OnChangeComplete fires once, when the drag or the key ends
  Info.Text := Format('%.0f to %.0f', [Price.Value, Price.EndValue]);
end;

function CreateFilters(aForm: TForm): TFilters;
begin
  Result := TFilters.Create(aForm);

  Result.Day := TsgcFMXDatePicker.Create(aForm);
  Result.Day.Parent := aForm;
  Result.Day.SetBounds(16, 16, 180, 32);
  Result.Day.Date := Date;

  // two thumbs, steps of 10, a tick and a label every 100
  Result.Price := TsgcFMXRangeSlider.Create(aForm);
```

```
  Result.Price.Parent := aForm;
  Result.Price.SetBounds(16, 64, 320, 56);
  Result.Price.Min := 0;
  Result.Price.Max := 500;
  Result.Price.RangeMode := True;
  Result.Price.Step := 10;
  Result.Price.TickInterval := 100;
  Result.Price.ShowLabels := True;
  Result.Price.Value := 100;
  Result.Price.EndValue := 300;
  Result.Price.OnChangeComplete := Result.PriceComplete;

  Result.Colour := TsgcFMXColorPicker.Create(aForm);
  Result.Colour.Parent := aForm;
  Result.Colour.SetBounds(16, 136, 240, 120);
  Result.Colour.Parts := [ecpSwatches, ecpHex];
  Result.Colour.SelectedColor := $FF2A78D6;

  Result.Info := TLabel.Create(aForm);
  Result.Info.Parent := aForm;
  Result.Info.SetBounds(16, 272, 320, 20);
  Result.PriceComplete(nil);
end;
```

Véase también

- Controles
- Plataformas
- DPI

Inicio rápido C++Builder

Instale los paquetes de sgcEditors para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de include y de biblioteca del proyecto, y use los componentes de la página de paleta **SGC Editors** exactamente como en Delphi. El instalador añade las dos rutas al IDE por usted. Los encabezados .hpp y los archivos .lib generados de sgcEditors y sgcUICore están en esa única carpeta.

Qué hay que saber en C++

- Incluya el encabezado de la unidad que declara un control: `sgcEditors_VCL_Date.hpp` para el calendario y los selectores, `sgcEditors_VCL_Value.hpp` para el control deslizante, la valoración y el spin edit, `sgcEditors_VCL_Color.hpp`, `sgcEditors_VCL_Token.hpp` y `sgcEditors_VCL_DB.hpp`. Las enumeraciones están en `sgcEditors_Types.hpp`.
- Los eventos son closures, de modo que sus manejadores son funciones miembro __fastcall de una clase. Un parámetro var como aAccept es una referencia bool.
- Date, MinDate y MaxDate son propiedades TDate. Asígneles un TDateTime, por ejemplo `Picker->Date = Date() + 1`.
- Las clases de modelo tienen constructores sin argumentos, como `new TsgcCalendarModel()`, y las destruye usted mismo con delete. Consulte Salida sin ventana.

En código

Véase también

- Inicio rápido VCL
- Inicio rápido data-aware
- Plataformas

Inicio rápido data-aware

Los controles data-aware enlazan un selector de fecha, un selector de hora, un spin edit, una valoración o un control deslizante de rango con un campo, como TDBEdit. Forman parte de las ediciones Registered y Trial, para VCL en sgcEditors_VCL_DB y para FireMonkey en sgcEditors_FMX_DB. FireMonkey no tiene TFieldDataLink, de modo que los dos frameworks comparten TsgcEditorFieldLink, un TDataLink de sgcEditors_Data.

En tiempo de diseño

1. Coloque un dataset y un **TDataSource** en un formulario.
2. Coloque un **TsgcVCLDBDatePicker** (o su gemelo FMX), asigne DataSource al origen de datos y elija el campo en DataField.
3. Para un **TsgcVCLDBRangeSlider** en modo de rango, asigne DataField para el tirador inferior y EndDataField para el tirador superior.

Cómo siguen los controles al campo

- El valor sigue al registro actual. Un campo nulo muestra un texto vacío en los selectores y en el spin edit, ningún glifo relleno en la valoración, y el tirador en Min en el control deslizante, con el tirador final en Max. AllowEmpty es True por defecto en los selectores y en el spin edit data-aware.
- Un cambio del usuario pone el dataset en modo de edición y marca el enlace como modificado. Asignar Value o Date en código también cuenta como un cambio del usuario. Cuando el dataset no puede entrar en modo de edición, vuelve el valor del campo.
- Enter y la pérdida de foco confirman el texto escrito y escriben el campo. Cuando el campo rechaza el valor, lanza una excepción y el foco se queda en el control, como en TDBEdit. Antes de un post, el dataset escribe un valor pendiente. Escape restaura el valor del campo.
- El selector de fecha conserva la parte de hora de un campo de fecha y hora, y el selector de hora conserva su parte de fecha. Un campo entero cambia el spin edit a esvInteger.
- Un control es de solo lectura mientras su ReadOnly es True, mientras no hay dataset, mientras el dataset o el campo es de solo lectura, para un campo calculado y, para el control deslizante en modo de rango, mientras el campo final no se puede modificar.

En código

El ejemplo crea un TClientDataSet y los cinco controles data-aware de VCL. Añada MidasLib a la cláusula uses del proyecto para que la aplicación no necesite midas.dll.

```
uses
  SysUtils, Classes, Controls, Forms, DB, DBClient,
  sgcEditors_VCL_DB;

function CreateBookings(aOwner: TComponent): TClientDataSet;
begin
  Result := TClientDataSet.Create(aOwner);
  Result.FieldDefs.Add('Guest', ftString, 40);
  Result.FieldDefs.Add('CheckIn', ftDate);
  Result.FieldDefs.Add('Arrival', ftTime);
  Result.FieldDefs.Add('Nights', ftInteger);
  Result.FieldDefs.Add('Score', ftFloat);
  Result.FieldDefs.Add('BudgetFrom', ftFloat);
  Result.FieldDefs.Add('BudgetTo', ftFloat);
  Result.CreateDataSet;
  Result.AppendRecord(['Ana', EncodeDate(2026, 10, 12),
    EncodeTime(15, 30, 0, 0), 3, 4, 200, 450]);
end;

procedure CreateDataEditors(aForm: TForm; aSource: TDataSource);
var
```

```
  oCheckIn: TsgcVCLDBDatePicker;
  oArrival: TsgcVCLDBTimePicker;
  oNights: TsgcVCLDBSpinEdit;
  oScore: TsgcVCLDBRating;
  oBudget: TsgcVCLDBRangeSlider;
begin
  oCheckIn := TsgcVCLDBDatePicker.Create(aForm);
  oCheckIn.Parent := aForm;
  oCheckIn.SetBounds(16, 16, 160, 28);
  oCheckIn.DataSource := aSource;
  oCheckIn.DataField := 'CheckIn';

  oArrival := TsgcVCLDBTimePicker.Create(aForm);
  oArrival.Parent := aForm;
  oArrival.SetBounds(192, 16, 120, 28);
  oArrival.ListInterval := 15;
  oArrival.DataSource := aSource;
  oArrival.DataField := 'Arrival';

  // an integer field switches the spin edit to esvInteger
  oNights := TsgcVCLDBSpinEdit.Create(aForm);
  oNights.Parent := aForm;
  oNights.SetBounds(16, 56, 120, 28);
  oNights.MinValue := 1;
  oNights.MaxValue := 30;
  oNights.Units := 'nights';
  oNights.DataSource := aSource;
  oNights.DataField := 'Nights';

  oScore := TsgcVCLDBRating.Create(aForm);
  oScore.Parent := aForm;
  oScore.SetBounds(16, 96, 160, 32);
  oScore.DataSource := aSource;
  oScore.DataField := 'Score';

  // in RangeMode, Value is DataField and EndValue is EndDataField
  oBudget := TsgcVCLDBRangeSlider.Create(aForm);
  oBudget.Parent := aForm;
  oBudget.SetBounds(16, 140, 320, 48);
  oBudget.Min := 0;
  oBudget.Max := 1000;
  oBudget.Step := 50;
  oBudget.RangeMode := True;
  oBudget.DataSource := aSource;
  oBudget.DataField := 'BudgetFrom';
  oBudget.EndDataField := 'BudgetTo';
end;
```

Véase también

- Controles
- Selector de fecha DB
- TsgcEditorFieldLink
- Ediciones
- Solución de problemas

Calendario mensual

Clases: TsgcVCLMonthCalendar (unidad sgcEditors_VCL_Date) y TsgcFMXMonthCalendar (unidad sgcEditors_FMX_Date)
El calendario mensual es un control dibujado a medida, que recibe el foco y no tiene edit. Muestra un mes con los nombres de los días y, opcionalmente, los números de semana, y se amplía a los 12 meses de un año y a los años de una década cuando el usuario hace clic en el título o pulsa Ctrl+Up. Selecciona una fecha o, con SelectionMode de esmRange, un rango de días. Los días fuera de MinDate y MaxDate, los DisabledDates, los DisabledWeekDays y los días que OnIsDateDisabled deshabilita se dibujan tachados y no se pueden seleccionar. El mismo modelo, TsgcCalendarModel, dibuja la lista desplegable de los selectores de fecha.
La clase VCL desciende de TsgcVCLCustomEditor y la clase FireMonkey de TsgcFMXCustomEditor. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Date	TDate	La fecha seleccionada en modo esmSingle, 0 cuando no hay ninguna seleccionada. El modo de rango no la modifica.
SelectionMode	TsgcEditorSelectionMode	esmSingle (por defecto) selecciona una fecha. esmRange selecciona un rango: el primer clic lo inicia, el segundo lo termina, y Shift con un clic o con las flechas lo extiende desde el primer día.
RangeStart, RangeEnd	TDate	El primer y el último día del rango, ambos incluidos.
MaxRangeDays	Integer	El máximo de días que puede contener un rango, con ambos extremos incluidos. 0 (por defecto) significa sin límite.
MinDate, MaxDate	TDate	El primer y el último día que el usuario puede elegir, comparados como días completos. 0 (por defecto) significa sin límite, y los límites invertidos se intercambian.
DisabledDates	TStrings	Días que no se pueden elegir, una fecha ISO (yyyy-mm-dd) por línea.
DisabledWeekDays	TsgcEditorWeekDays	Días de la semana que no se pueden elegir, un conjunto de ewdMonday a ewdSunday. Por defecto [].
FirstDayOfWeek	TsgcEditorFirstDayOfWeek	efdLocale (por defecto) toma el primer día de la semana del sistema operativo, el lunes cuando no tiene ninguno. efdMonday, efdSaturday y efdSunday lo fuerzan.
ShowWeekNumbers	Boolean	Añade una columna con el número de semana de cada fila. Por defecto False.
WeekNumbering	TsgcEditorWeekNumbering	ewnISO8601 (por defecto): la semana 1 contiene el primer jueves del año. ewnSimple: la semana 1 contiene el 1 de enero.
ShowToday	Boolean	Marca el día actual. Por defecto True.
ShowOtherMonthDays	Boolean	Muestra los días del mes anterior y del siguiente que completan las seis semanas de la cuadrícula. Por defecto True.

View	TsgcEditorCalendarView	ecvMonth (por defecto), ecvYear (los 12 meses de un año) o ecvDecade (los años de una década).
DisplayMonth	TDate	El primer día del mes mostrado, que es siempre el mes del foco del teclado. No se guarda en el formulario.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
FormatSettings	TFormatSettings	Public. Los nombres, formatos y separadores que usa el control. Empieza con la configuración regional del usuario. Consulte Localización.
CalendarModel	TsgcCalendarModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
GoToDate(aDate)		Muestra el mes de aDate, mantenido dentro de los límites, y mueve el foco del teclado a él. No selecciona la fecha.
IsDateEnabled(aDate)	Boolean	True cuando la fecha está dentro de los límites, no cae en un día de la semana deshabilitado, no está en DisabledDates y no la deshabilita OnIsDateDisabled.
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	La fecha seleccionada o el rango cambió.
OnSelect	TNotifyEvent	Se elige una fecha en modo esmSingle, con el ratón o con Enter o Espacio.
OnRangeSelect	TNotifyEvent	Un rango está completo: el segundo clic, o un clic o una flecha con Shift.

OnIsDate-Disabled	TsgcCalen-darIsDateDi-sabledEvent	(Sender; const aDate: TDateTime; var aDisabled: Boolean). Asigne True a aDisabled para deshabilitar un día, como un festivo procedente de una base de datos. Se consulta para los días que muestra el calendario y para el día que se está seleccionando, así que debe ser rápido.
OnView-Change	TNotifyEvent	Vista cambiada: el usuario amplió o redujo.
OnDis-play-Month-Change	TNotifyEvent	Se muestra otro mes.

Teclado

Las flechas mueven el foco un día o una semana, PageUp y PageDown un mes y, con Ctrl, un año. Ctrl+Up reduce. Enter o Espacio seleccionan el día con foco, o amplían en las vistas de año y de década. Shift con las flechas extiende un rango. Los días deshabilitados se omiten.

Ejemplo: una estancia de hasta dos semanas

```
uses
  SysUtils, DateUtils, Classes, Controls, StdCtrls,
  sgcEditors_Types, sgcEditors_VCL_Date;

type
  TStayEvents = class
  public
    Calendar: TsgcVCLMonthCalendar;
    Summary: TLabel;
    procedure RangeSelect(Sender: TObject);
    procedure IsDateDisabled(Sender: TObject; const aDate: TDateTime;
      var aDisabled: Boolean);
  end;

procedure TStayEvents.RangeSelect(Sender: TObject);
begin
  // RangeStart and RangeEnd are whole days, both included
  Summary.Caption := DateToStr(Calendar.RangeStart) + ' to ' +
    DateToStr(Calendar.RangeEnd) + ', ' +
    IntToStr(DaysBetween(Calendar.RangeEnd, Calendar.RangeStart)) + ' nights';
end;

procedure TStayEvents.IsDateDisabled(Sender: TObject; const aDate: TDateTime;
  var aDisabled: Boolean);
var
  vYear, vMonth, vDay: Word;
begin
  // the hotel is closed on the first day of every month
  DecodeDate(aDate, vYear, vMonth, vDay);
  aDisabled := vDay = 1;
end;

procedure SetupStay(aCalendar: TsgcVCLMonthCalendar; aEvents: TStayEvents);
begin
  aCalendar.SelectionMode := esmRange;
  aCalendar.MaxRangeDays := 14;
  aCalendar.MinDate := Date;
  aCalendar.MaxDate := IncMonth(Date, 12);
  aCalendar.FirstDayOfWeek := efdMonday;
  aCalendar.ShowWeekNumbers := True;
  aCalendar.DisabledWeekDays := [ewdSunday];
  // one ISO date per line
```

```
  aCalendar.DisabledDates.Add('2026-12-25');
  aCalendar.DisabledDates.Add('2027-01-06');
  aEvents.Calendar := aCalendar;
  aCalendar.OnRangeSelect := aEvents.RangeSelect;
  aCalendar.OnIsDateDisabled := aEvents.IsDateDisabled;
  aCalendar.GoToDate(Date);
end;
```

Véase también

- Selector de fecha
- Selector de fecha y hora
- Localización
- TsgcEditorLocale

Selector de fecha

Clases: TsgcVCLDatePicker (unidad sgcEditors_VCL_Date) y TsgcFMXDatePicker (unidad sgcEditors_FMX_Date)
El selector de fecha es un edit nativo para una fecha, con un calendario desplegable. El usuario escribe la fecha, cambia el día, el mes o el año bajo el cursor con Up y Down, o la elige en el calendario. Comparte sus clases base, TsgcVCLCustomDatePicker y TsgcFMXCustomDatePicker, y su modelo, TsgcDatePickerModel, con el selector de hora y el selector de fecha y hora.
La clase VCL desciende de TsgcVCLCustomDatePicker y la clase FireMonkey de TsgcFMXCustomDatePicker. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Date	TDate	La fecha. Lea IsEmpty primero cuando AllowEmpty es True.
DisplayFormat	string	El formato del texto, como dd/mm/yyyy o ddd d mmm yyyy. Vacío (por defecto) usa el formato de fecha corta de FormatSettings. Consulte Entrada de texto e IME.
AllowEmpty	Boolean	Permite al usuario borrar el texto, lo que vacía el valor. Por defecto False, y entonces un texto vacío devuelve el último valor.
IsEmpty	Boolean	Public, solo lectura. True mientras el valor está vacío.
TextHint	string	La pista gris de un edit vacío. VCL la muestra desde Delphi 2009, y FireMonkey la muestra como el TextPrompt del edit.
DroppedDown	Boolean	Public. True mientras la lista desplegable está abierta. Asignarlo la abre o la cierra.
MinDate, MaxDate	TDate	El primer y el último día que el usuario puede elegir, comparados como días completos. 0 (por defecto) significa sin límite, y los límites invertidos se intercambian.
DisabledDates	TStrings	Días que no se pueden elegir, una fecha ISO (yyyy-mm-dd) por línea.
DisabledWeekDays	TsgcEditorWeekDays	Días de la semana que no se pueden elegir, un conjunto de ewdMonday a ewdSunday. Por defecto [].
FirstDayOfWeek, ShowWeekNumbers		Como en el calendario mensual, para el calendario desplegable.
DropDownCalendar	TsgcCalendarModel	Public, solo lectura. El modelo del calendario desplegable.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.

TextStyle	TsgcUI-TextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
FormatSettings	TFormatSettings	Public. Los nombres, formatos y separadores que usa el control. Empieza con la configuración regional del usuario. Consulte Localización.
PickerModel	TsgcDatePickerModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
Clear		Vacía el valor cuando AllowEmpty es True. En caso contrario, no hace nada.
DropDown		Abre la lista desplegable, como el botón, F4 y Alt+Down.
CloseUp(aAccept)		Cierra la lista desplegable. aAccept True confirma el texto, False restaura el valor que tenía al abrirse.
IsDateEnabled(aDate)	Boolean	True cuando la fecha está dentro de los límites y no está deshabilitada en el calendario desplegable. Una confirmación rechaza las fechas para las que es False.
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Un valor nuevo: escrito y confirmado, elegido en la lista desplegable o cambiado con Up y Down.
OnDropDown, OnCloseUp	TNotifyEvent	La lista desplegable se abre o se cierra.
OnParseError	TsgcDatePickerParseErrorEvent	(Sender; const aText: string; var aValue: TDateTime; var aAccept: Boolean). El texto confirmado no se puede interpretar. aValue llega como el valor actual y aAccept como False. Asigne True a aAccept y un valor a aValue para tomar otro valor, que se mantiene dentro de los límites. En caso contrario vuelve el último valor válido.

OnIsDate-Disabled	TsgcCalen-darIsDate-DisabledE-vent	Como en el calendario mensual: deshabilita un día en la lista desplegable y al confirmar.

Escritura y lista desplegable

- El texto se escribe en un edit nativo, de modo que IME, el portapapeles y deshacer funcionan. La interpretación es tolerante y sigue los nombres y separadores de FormatSettings. Consulte Entrada de texto e IME.
- Up y Down confirman el texto escrito y después cambian el campo bajo el cursor, que da la vuelta dentro de su rango, y vuelven a seleccionar el campo. Los límites se aplican después del paso.
- Enter, la pérdida de foco y una elección en la lista desplegable confirman el texto. Un valor fuera de los límites se mueve dentro de ellos, y una fecha deshabilitada se rechaza.
- F4, Alt+Down y el botón abren la lista desplegable en el valor actual. Mientras está abierta, sigue el texto escrito, Up, Down, PageUp y PageDown van hacia ella, y Escape la cierra y restaura el valor y el texto que tenía al abrirse.

Ejemplo: una fecha de vencimiento que entiende "hoy"

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Date;

type
  TDueEvents = class
  public
    procedure ParseError(Sender: TObject; const aText: string;
      var aValue: TDateTime; var aAccept: Boolean);
  end;

procedure TDueEvents.ParseError(Sender: TObject; const aText: string;
  var aValue: TDateTime; var aAccept: Boolean);
begin
  // words the parse does not know; the value is clamped to the limits
  if SameText(Trim(aText), 'today') then
  begin
    aValue := Date;
    aAccept := True;
  end
  else if SameText(Trim(aText), 'tomorrow') then
  begin
    aValue := Date + 1;
    aAccept := True;
  end;
end;

procedure SetupDueDate(aPicker: TsgcVCLDatePicker; aEvents: TDueEvents);
begin
  aPicker.DisplayFormat := 'dd mmm yyyy';
  aPicker.AllowEmpty := True;
  aPicker.TextHint := 'No due date';
  aPicker.MinDate := Date;
  aPicker.DisabledWeekDays := [ewdSaturday, ewdSunday];
  aPicker.ShowWeekNumbers := True;
  aPicker.OnParseError := aEvents.ParseError;
  aPicker.Clear;
end;

function DueDateText(aPicker: TsgcVCLDatePicker): string;
begin
  if aPicker.IsEmpty then
    Result := 'none'
  else
```

```
    Result := FormatDateTime('yyyy-mm-dd', aPicker.Date);
end;
```

Véase también

- Calendario mensual
- Selector de fecha y hora
- Selector de fecha DB
- Entrada de texto e IME

Selector de hora

Clases: TsgcVCLTimePicker (unidad sgcEditors_VCL_Date) y TsgcFMXTimePicker (unidad sgcEditors_FMX_Date)
El selector de hora es un edit nativo para una hora del día, con una lista desplegable de horas. El usuario escribe la hora, cambia la hora, el minuto, el segundo o AM y PM bajo el cursor con Up y Down, o elige una hora en la lista. La lista empieza en 00:00 y contiene una hora cada ListInterval minutos entre MinTime y MaxTime, escrita con los campos de hora de DisplayFormat.
La clase VCL desciende de TsgcVCLCustomDatePicker y la clase FireMonkey de TsgcFMXCustomDatePicker. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Time	TTime	La hora del día. El selector de hora empieza en 00:00.
DisplayFormat	string	El formato del texto, como hh:nn o h:nn ampm. Un formato con ss muestra los segundos. Vacío (por defecto) usa el formato de hora corta de FormatSettings.
MinTime, MaxTime	TTime	La hora más temprana y la más tardía que puede asignar el usuario. 0 (por defecto) significa sin límite en ese lado.
ListInterval	Integer	Los minutos entre dos horas de la lista desplegable. Por defecto 30. 0 elimina la lista desplegable y su botón.
AllowEmpty	Boolean	Permite al usuario borrar el texto, lo que vacía el valor. Por defecto False, y entonces un texto vacío devuelve el último valor.
IsEmpty	Boolean	Public, solo lectura. True mientras el valor está vacío.
TextHint	string	La pista gris de un edit vacío. VCL la muestra desde Delphi 2009, y FireMonkey la muestra como el TextPrompt del edit.
DroppedDown	Boolean	Public. True mientras la lista desplegable está abierta. Asignarlo la abre o la cierra.
DropDownTimeList	TsgcEditorListModel	Public, solo lectura. El modelo de la lista desplegable.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.

FormatSettings	TFormatSettings	Public. Los nombres, formatos y separadores que usa el control. Empieza con la configuración regional del usuario. Consulte Localización.
PickerModel	TsgcDatePickerModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
Clear		Vacía el valor cuando AllowEmpty es True. En caso contrario, no hace nada.
DropDown		Abre la lista desplegable, como el botón, F4 y Alt+Down.
CloseUp(aAccept)		Cierra la lista desplegable. aAccept True confirma el texto, False restaura el valor que tenía al abrirse.
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Un valor nuevo: escrito y confirmado, elegido en la lista desplegable o cambiado con Up y Down.
OnDropDown, OnCloseUp	TNotifyEvent	La lista desplegable se abre o se cierra.
OnParseError	TsgcDatePickerParseErrorEvent	(Sender; const aText: string; var aValue: TDateTime; var aAccept: Boolean). El texto confirmado no se puede interpretar. aValue llega como el valor actual y aAccept como False. Asigne True a aAccept y un valor a aValue para tomar otro valor, que se mantiene dentro de los límites. En caso contrario vuelve el último valor válido.

Escritura y lista desplegable

- El texto se escribe en un edit nativo, de modo que IME, el portapapeles y deshacer funcionan. La interpretación es tolerante y sigue los nombres y separadores de FormatSettings. Consulte Entrada de texto e IME.
- Up y Down confirman el texto escrito y después cambian el campo bajo el cursor, que da la vuelta dentro de su rango, y vuelven a seleccionar el campo. Los límites se aplican después del paso.
- Enter, la pérdida de foco y una elección en la lista desplegable confirman el texto. Un valor fuera de los límites se mueve dentro de ellos, y una fecha deshabilitada se rechaza.
- F4, Alt+Down y el botón abren la lista desplegable en el valor actual. Mientras está abierta, sigue el texto escrito, Up, Down, PageUp y PageDown van hacia ella, y Escape la cierra y restaura el valor y el texto que tenía al abrirse.

Ejemplo: una reunión en horario de oficina

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_VCL_Date;

type
  TMeetingEvents = class
  public
    Start: TsgcVCLTimePicker;
    Finish: TsgcVCLTimePicker;
    procedure StartChange(Sender: TObject);
  end;

procedure TMeetingEvents.StartChange(Sender: TObject);
begin
  // the meeting ends at least 30 minutes after it starts
  Finish.MinTime := Start.Time + EncodeTime(0, 30, 0, 0);
  if Finish.Time < Finish.MinTime then
    Finish.Time := Finish.MinTime;
end;

procedure SetupMeeting(aEvents: TMeetingEvents);
begin
  // 24 hour text, a time list every 15 minutes, office hours only
  aEvents.Start.DisplayFormat := 'hh:nn';
  aEvents.Start.ListInterval := 15;
  aEvents.Start.MinTime := EncodeTime(8, 0, 0, 0);
  aEvents.Start.MaxTime := EncodeTime(18, 0, 0, 0);
  aEvents.Start.Time := EncodeTime(9, 0, 0, 0);
  aEvents.Finish.DisplayFormat := 'hh:nn';
  aEvents.Finish.ListInterval := 15;
  aEvents.Finish.MaxTime := EncodeTime(19, 0, 0, 0);
  aEvents.Start.OnChange := aEvents.StartChange;
  aEvents.StartChange(nil);
end;
```

Véase también

- Selector de fecha
- Selector de fecha y hora
- Selector de hora DB

Selector de fecha y hora

Clases: TsgcVCLDateTimePicker (unidad sgcEditors_VCL_Date) y TsgcFMXDateTimePicker (unidad sgcEditors_FMX_Date)
El selector de fecha y hora edita una fecha y una hora en un solo texto. Su lista desplegable muestra el calendario y la lista de horas una al lado de la otra. Una elección en el calendario mantiene abierta la lista, para que el usuario elija después la hora, y una elección en la lista de horas la cierra. MinDate, MaxDate y los días deshabilitados limitan la fecha, y MinTime y MaxTime limitan la hora del día.
La clase VCL desciende de TsgcVCLCustomDatePicker y la clase FireMonkey de TsgcFMXCustomDatePicker. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
DateTime	TDateTime	La fecha y la hora.
DisplayFormat	string	El formato del texto, como dd/mm/yyyy hh:nn. Vacío (por defecto) usa la fecha corta y la hora corta de FormatSettings, separadas por un espacio.
MinDate, MaxDate	TDate	El primer y el último día que el usuario puede elegir, comparados como días completos. 0 (por defecto) significa sin límite, y los límites invertidos se intercambian.
DisabledDates	TStrings	Días que no se pueden elegir, una fecha ISO (yyyy-mm-dd) por línea.
DisabledWeekDays	TsgcEditorWeekDays	Días de la semana que no se pueden elegir, un conjunto de ewdMonday a ewdSunday. Por defecto [].
MinTime, MaxTime	TTime	La hora del día más temprana y la más tardía. 0 (por defecto) significa sin límite en ese lado.
ListInterval	Integer	Los minutos entre dos horas de la lista de horas. Por defecto 30.
FirstDayOfWeek, ShowWeekNumbers		Como en el calendario mensual, para el calendario desplegable.
AllowEmpty	Boolean	Permite al usuario borrar el texto, lo que vacía el valor. Por defecto False, y entonces un texto vacío devuelve el último valor.
IsEmpty	Boolean	Public, solo lectura. True mientras el valor está vacío.
TextHint	string	La pista gris de un edit vacío. VCL la muestra desde Delphi 2009, y FireMonkey la muestra como el TextPrompt del edit.
DroppedDown	Boolean	Public. True mientras la lista desplegable está abierta. Asignarlo la abre o la cierra.
DropDownCalendar, DropDownTimeList	TsgcCalendarModel, TsgcEditorListModel	Public, solo lectura. Los modelos de las dos partes de la lista desplegable.

Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
FormatSettings	TFormatSettings	Public. Los nombres, formatos y separadores que usa el control. Empieza con la configuración regional del usuario. Consulte Localización.
PickerModel	TsgcDatePickerModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
Clear		Vacía el valor cuando AllowEmpty es True. En caso contrario, no hace nada.
DropDown		Abre la lista desplegable, como el botón, F4 y Alt+Down.
CloseUp(aAccept)		Cierra la lista desplegable. aAccept True confirma el texto, False restaura el valor que tenía al abrirse.
IsDateEnabled(aDate)	Boolean	True cuando la fecha está dentro de los límites y no está deshabilitada en el calendario desplegable.
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Un valor nuevo: escrito y confirmado, elegido en la lista desplegable o cambiado con Up y Down.
OnDropDown, OnCloseUp	TNotifyEvent	La lista desplegable se abre o se cierra.

OnParseError	TsgcDatePickerParseErrorEvent	(Sender; const aText: string; var aValue: TDateTime; var aAccept: Boolean). El texto confirmado no se puede interpretar. aValue llega como el valor actual y aAccept como False. Asigne True a aAccept y un valor a aValue para tomar otro valor, que se mantiene dentro de los límites. En caso contrario vuelve el último valor válido.
OnIsDateDisabled	TsgcCalendarIsDateDisabledEvent	Como en el calendario mensual.

Escritura y lista desplegable

- El texto se escribe en un edit nativo, de modo que IME, el portapapeles y deshacer funcionan. La interpretación es tolerante y sigue los nombres y separadores de FormatSettings. Consulte Entrada de texto e IME.
- Up y Down confirman el texto escrito y después cambian el campo bajo el cursor, que da la vuelta dentro de su rango, y vuelven a seleccionar el campo. Los límites se aplican después del paso.
- Enter, la pérdida de foco y una elección en la lista desplegable confirman el texto. Un valor fuera de los límites se mueve dentro de ellos, y una fecha deshabilitada se rechaza.
- F4, Alt+Down y el botón abren la lista desplegable en el valor actual. Mientras está abierta, sigue el texto escrito, Up, Down, PageUp y PageDown van hacia ella, y Escape la cierra y restaura el valor y el texto que tenía al abrirse.

En la lista desplegable del selector de fecha y hora, Up, Down, PageUp y PageDown van al calendario cuando el cursor está en un campo de fecha y a la lista de horas cuando está en un campo de hora.

Ejemplo: una franja de entrega

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Date;

procedure SetupDelivery(aPicker: TsgcVCLDateTimePicker);
begin
  // the date and the time in one text; the drop-down shows the calendar
  // and the time list side by side
  aPicker.DisplayFormat := 'ddd dd/mm/yyyy hh:nn';
  aPicker.MinDate := Date + 1;
  aPicker.MaxDate := Date + 60;
  aPicker.DisabledWeekDays := [ewdSunday];
  aPicker.MinTime := EncodeTime(9, 0, 0, 0);
  aPicker.MaxTime := EncodeTime(17, 0, 0, 0);
  aPicker.ListInterval := 60;
  aPicker.DateTime := Date + 1 + EncodeTime(9, 0, 0, 0);
end;

function DeliveryWindow(aPicker: TsgcVCLDateTimePicker): string;
begin
  // DateTime holds both parts
  Result := FormatDateTime('dddd d mmmm, hh:nn', aPicker.DateTime) + ' to ' +
    FormatDateTime('hh:nn', aPicker.DateTime + EncodeTime(1, 0, 0, 0));
end;
```

Véase también

- Selector de fecha
- Selector de hora
- Calendario mensual

Control deslizante de rango

Clases: TsgcVCLRangeSlider (unidad sgcEditors_VCL_Value) y TsgcFMXRangeSlider (unidad sgcEditors_FMX_Value)
El control deslizante de rango es un control dibujado a medida, que recibe el foco y elige un número entre Min y Max con un tirador, o un rango con dos tiradores en RangeMode. Es horizontal, con Min a la izquierda, o vertical, con Min abajo. Puede dibujar marcas, etiquetas y una sugerencia con el valor sobre el tirador arrastrado. Un clic en el carril mueve hasta allí el tirador más cercano y lo arrastra, y los tiradores nunca se cruzan.
RangeMode requiere la edición Registered o Trial. En Community la propiedad se conserva, se dibuja y se edita un tirador, y EndValue se conserva pero no se muestra. Consulte Ediciones.
La clase VCL desciende de TsgcVCLCustomEditor y la clase FireMonkey de TsgcFMXCustomEditor. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Min, Max	Double	Los extremos de la escala. Por defecto 0 y 100.
Value	Double	El valor, o el extremo inferior del rango en RangeMode. Por defecto 0.
EndValue	Double	El extremo superior del rango en RangeMode. Por defecto 100.
RangeMode	Boolean	Dos tiradores, Value y EndValue. Por defecto False. Ediciones Registered y Trial.
MinRange	Double	La menor distancia entre los dos tiradores. Por defecto 0.
Step	Double	Cada valor se ajusta a Min más un múltiplo de Step, y las flechas mueven un Step. Por defecto 1. 0 es continuo, y entonces las flechas mueven un 1% de la escala.
LargeStep	Double	El paso de PageUp y PageDown. Por defecto 10. 0 es una décima parte de la escala.
Orientation	TsgcEditorOrientation	eorHorizontal (por defecto) o eorVertical.
TickInterval	Double	La distancia entre dos marcas. 0 (por defecto) no dibuja marcas.
TickPlacement	TsgcEditorTickPlacement	etpBottomRight (por defecto), etpTopLeft, etpBoth o etpNone.
ShowLabels	Boolean	Escribe el valor en cada marca, o en Min y Max cuando no hay marcas. Las etiquetas que se solaparían se omiten. Por defecto False.
LabelFormat	string	Una máscara FormatFloat para las etiquetas y la sugerencia de valor. Vacío (por defecto) escribe tantos decimales como necesita Step.
ShowValueTip	Boolean	Muestra el valor sobre el tirador mientras se arrastra. Por defecto True.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.

AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
FormatSettings	TFormatSettings	Public. Los nombres, formatos y separadores que usa el control. Empieza con la configuración regional del usuario. Consulte Localización.
SliderModel	TsgcSliderModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Value o EndValue cambió. Se dispara durante un arrastre, por cada valor nuevo.
OnChangeComplete	TNotifyEvent	El usuario terminó un cambio: soltar el ratón, soltar la tecla o un paso de la rueda. Úselo para trabajo que no debe ejecutarse durante un arrastre, como una consulta.
OnFormatLabel	TsgcSliderFormatLabelEvent	(Sender; aValue: Double; var aText: string). Escribe el texto de una etiqueta y de la sugerencia de valor, como una moneda o una unidad.

Teclado

Las flechas mueven el tirador activo un Step, PageUp y PageDown un LargeStep, y Home y End van a los extremos. En RangeMode, Tab pasa del tirador inferior al superior y Shift+Tab vuelve, y luego el siguiente Tab sale del control. Ctrl+Left o Ctrl+Down activa el tirador inferior, y Ctrl+Right o Ctrl+Up el superior. En FireMonkey, Tab siempre pasa al control siguiente, así que las teclas Ctrl cambian de tirador.

Ejemplo: un filtro de precio

```
uses
  SysUtils, Classes, Controls, StdCtrls,
  sgcEditors_Types, sgcEditors_VCL_Value;
```

```
type
  TPriceEvents = class
  public
    Slider: TsgcVCLRangeSlider;
    Summary: TLabel;
    procedure FormatLabel(Sender: TObject; aValue: Double; var aText: string);
    procedure ChangeComplete(Sender: TObject);
  end;

procedure TPriceEvents.FormatLabel(Sender: TObject; aValue: Double;
  var aText: string);
begin
  // the text of the labels and of the value tip
  aText := FormatFloat('0', aValue) + ' EUR';
end;

procedure TPriceEvents.ChangeComplete(Sender: TObject);
begin
  // once on mouse up, key up or wheel, not on every step of a drag
  Summary.Caption := 'From ' + FormatFloat('0', Slider.Value) + ' to ' +
    FormatFloat('0', Slider.EndValue) + ' EUR';
end;

procedure SetupPrice(aEvents: TPriceEvents);
begin
  aEvents.Slider.Min := 0;
  aEvents.Slider.Max := 500;
  aEvents.Slider.Step := 10;
  aEvents.Slider.LargeStep := 50;
  // two thumbs at least 50 apart (Registered and Trial editions)
  aEvents.Slider.RangeMode := True;
  aEvents.Slider.MinRange := 50;
  aEvents.Slider.Value := 100;
  aEvents.Slider.EndValue := 300;
  aEvents.Slider.TickInterval := 100;
  aEvents.Slider.TickPlacement := etpBottomRight;
  aEvents.Slider.ShowLabels := True;
  aEvents.Slider.OnFormatLabel := aEvents.FormatLabel;
  aEvents.Slider.OnChangeComplete := aEvents.ChangeComplete;
end;
```

Véase también

- Control deslizante de rango DB
- Spin edit
- Teclado y accesibilidad

Selector de color

Clases: TsgcVCLColorPicker (unidad sgcEditors_VCL_Color) y TsgcFMXColorPicker (unidad sgcEditors_FMX_Color)
El selector de color muestra, de arriba abajo, una cuadrícula de muestras, un cuadrado de saturación y valor, una barra de tono, una barra alfa y una fila con una vista previa y un edit nativo para el texto hexadecimal. Parts elige cuáles se muestran. El edit hexadecimal acepta textos #RGB, #RGBA, #RRGGBB y #RRGGBBAA. El selector guarda el tono aparte del color, de modo que un gris conserva su tono y el cuadrado no salta.
El cuadrado HSV y la barra alfa requieren la edición Registered o Trial. En Community no se dibujan, el alfa es siempre $FF y el texto hexadecimal es #RRGGBB. Consulte Ediciones.
La clase VCL desciende de TsgcVCLCustomEditor y la clase FireMonkey de TsgcFMXCustomEditor. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
SelectedColor	TsgcUIColor	El color como ARGB, por ejemplo $FF2A78D6. Tiene la disposición del TAlphaColor de FireMonkey.
SelectedVCLColor	TColor	Solo VCL, public. El color como un TColor opaco, para lectura y escritura.
Parts	TsgcEditorColorParts	Las partes mostradas: ecpSwatches, ecpHSV, ecpAlpha y ecpHex. Por defecto las cuatro.
Swatches	TStrings	Los colores de las muestras, un color hexadecimal como #1E88E5 por línea. Por defecto 24 colores de las paletas de sgcUICore y grises.
SwatchSize	Double	El tamaño de una muestra en píxeles lógicos. Por defecto 20.
SwatchColumns	Integer	Las muestras de una fila. Por defecto 8.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
ColorModel	TsgcColorPickerModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El color cambió. Se dispara durante un arrastre en el cuadrado o en las barras, por cada color nuevo.
OnSelect	TNotifyEvent	Se elige un color: un clic en una muestra, soltar el ratón tras un arrastre, Enter, o un texto hexadecimal confirmado con Enter o al perder el foco.

Teclado

- En VCL el control tiene dos paradas de foco. Tab recorre las partes, después entra en el edit hexadecimal, y el siguiente Tab sale. En solo lectura, el control es una sola parada.
- En FireMonkey el control es una sola parada de tabulación con dos paradas internas: Ctrl+flechas pasan de las partes al edit hexadecimal y viceversa.
- En las muestras, las flechas, Home y End mueven el foco y Enter o Espacio eligen la muestra. En el cuadrado, Left y Right cambian la saturación y Up y Down el valor en un 1%. En las barras, las flechas se mueven un grado o un 1%. Shift avanza diez veces más.
- Un texto hexadecimal que no es un color vuelve a mostrar el color, y Escape lo restaura.

Ejemplo: colores de marca para una etiqueta

```
uses
  SysUtils, Classes, Controls, Graphics, ExtCtrls,
  sgcEditors_Types, sgcEditors_VCL_Color;

type
  TLabelColorEvents = class
  public
    Picker: TsgcVCLColorPicker;
    Preview: TPanel;
    procedure ColorSelect(Sender: TObject);
  end;

procedure TLabelColorEvents.ColorSelect(Sender: TObject);
begin
  // a swatch click, Enter, the hex edit or the end of a drag
  Preview.Color := Picker.SelectedVCLColor;
end;

procedure SetupLabelColor(aEvents: TLabelColorEvents);
begin
  // six brand colours in one row and the hex edit, no square and no alpha
  aEvents.Picker.Parts := [ecpSwatches, ecpHex];
  aEvents.Picker.Swatches.Clear;
  aEvents.Picker.Swatches.Add('#1E88E5');
```

```
  aEvents.Picker.Swatches.Add('#43A047');
  aEvents.Picker.Swatches.Add('#FDD835');
  aEvents.Picker.Swatches.Add('#FB8C00');
  aEvents.Picker.Swatches.Add('#E53935');
  aEvents.Picker.Swatches.Add('#8E24AA');
  aEvents.Picker.SwatchColumns := 6;
  aEvents.Picker.SwatchSize := 28;
  aEvents.Picker.SelectedVCLColor := clNavy;
  aEvents.Picker.OnSelect := aEvents.ColorSelect;
end;
```

Ejemplo: un color de relleno en FireMonkey

```
uses
  System.SysUtils, System.Classes, System.UITypes, FMX.Objects,
  sgcEditors_Types, sgcEditors_FMX_Color;

type
  TFillEvents = class
  public
    Picker: TsgcFMXColorPicker;
    Preview: TRectangle;
    procedure ColorChange(Sender: TObject);
  end;

procedure TFillEvents.ColorChange(Sender: TObject);
begin
  // SelectedColor is ARGB, the layout of TAlphaColor; OnChange follows
  // a drag on the square and the bars
  Preview.Fill.Color := TAlphaColor(Picker.SelectedColor);
end;

procedure SetupFill(aEvents: TFillEvents);
begin
  aEvents.Picker.Parts := [ecpSwatches, ecpHSV, ecpAlpha, ecpHex];
  aEvents.Picker.SelectedColor := $802A78D6;
  aEvents.Picker.OnChange := aEvents.ColorChange;
end;
```

Véase también

- Token edit
- Temas
- Ediciones

Token edit

Clases: TsgcVCLTokenEdit (unidad sgcEditors_VCL_Token) y TsgcFMXTokenEdit (unidad sgcEditors_FMX_Token)
El token edit contiene una lista de tokens, como etiquetas o direcciones de correo, dibujados como chips con un botón de quitar. Los chips fluyen de izquierda a derecha, y un edit nativo después del último chip recibe el siguiente token, de modo que IME, el portapapeles y deshacer funcionan. Un delimitador, Enter, Tab y la pérdida de foco convierten el texto escrito en un chip, y pegar a, b, c crea tres chips. Cada token se recorta, se comprueba con OnValidateToken, se comprueba si está duplicado y se cuenta contra MaxTokens. Cuando los chips no caben, el control se desplaza en vertical.
La lista de autocompletado requiere la edición Registered o Trial. En Community las propiedades se conservan y la lista nunca se abre. Consulte Ediciones.
La clase VCL desciende de TsgcVCLCustomEditor y la clase FireMonkey de TsgcFMXCustomEditor. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Tokens	TStrings	Los tokens, en orden. Cambiar la lista cambia los chips.
Delimiters	string	Los caracteres que terminan un token. Por defecto ,; y los saltos de línea siempre separan. Vacío deja solo Enter y los saltos de línea.
AllowDuplicates	Boolean	Añade un token que ya está en la lista. Por defecto False.
CaseSensitive	Boolean	Compara los tokens distinguiendo mayúsculas de minúsculas al buscar duplicados. Por defecto False.
MaxTokens	Integer	El máximo de tokens que admite la lista. 0 (por defecto) significa sin límite. Un texto que supera el límite se queda en el edit.
InvalidTokens	TsgcEditorInvalidTokens	etiReject (por defecto) mantiene en el edit un texto rechazado. etiMark lo añade como un chip marcado; TokenModel.IsTokenInvalid(aIndex) indica cuál.
TextHint	string	La pista gris del edit vacío.
AutoCompleteItems	TStrings	Los candidatos de la lista de autocompletado. Ediciones Registered y Trial.
AutoCompleteMatch	TsgcEditorMatchMode	emmStartsWith (por defecto) o emmContains.
AutoCompleteMinChars	Integer	Los caracteres que escribe el usuario antes de que se abra la lista. Por defecto 1.
AutoCompleteMaxItems	Integer	El máximo de filas que muestra la lista. 0 (por defecto) significa sin límite.
DroppedDown	Boolean	Public. True mientras la lista de autocompletado está abierta.

Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
TokenModel	TsgcTokenModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
AddToken(aText)	Boolean	Añade un token sin dividirlo, pasando por el recorte, OnValidateToken, la comprobación de duplicados y MaxTokens. True cuando se añadió.
RemoveToken(aIndex)	Boolean	Quita el token en aIndex. True cuando se quitó.
Clear		Quita todos los tokens y el texto escrito.
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	La lista de tokens cambió.
OnValidateToken	TsgcTokenValidateEvent	(Sender; var aText: string; var aValid: Boolean). aText llega recortado y se puede cambiar, por ejemplo para normalizarlo. Asigne False a aValid para rechazar el token, o para marcarlo con etiMark.
OnTokenAdded, OnTokenRemoved	TsgcTokenEvent	(Sender; aIndex: Integer; const aText: string). Se añadió o se quitó un chip.
OnAutoComplete	TsgcTokenAutoCompleteEvent	(Sender; const aText: string; aItems: TStrings). aItems son los candidatos de la lista. Cámbielos según el texto escrito, por ejemplo desde una base de datos, y la lista los filtra después. Ediciones Registered y Trial.

Teclado

Retroceso o Left en un edit vacío da el foco al último chip, y un segundo Retroceso lo quita. Left y Right recorren los chips, Right en el último vuelve al edit, Delete quita el chip con foco, y Escape o escribir vuelve al edit. Mientras la lista de autocompletado está abierta, Up, Down, PageUp y PageDown se mueven por ella, Enter elige la fila resaltada y Escape la cierra.

Ejemplo: destinatarios de correo

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Token;

type
  TRecipientEvents = class
  public
    procedure ValidateToken(Sender: TObject; var aText: string;
      var aValid: Boolean);
  end;

procedure TRecipientEvents.ValidateToken(Sender: TObject; var aText: string;
  var aValid: Boolean);
begin
  // aText comes trimmed; store addresses in lower case and mark the ones
  // without an @
  aText := LowerCase(aText);
  aValid := Pos('@', aText) > 1;
end;

procedure SetupRecipients(aTokens: TsgcVCLTokenEdit;
  aEvents: TRecipientEvents);
begin
  aTokens.Delimiters := ',; ';
  aTokens.MaxTokens := 10;
  aTokens.InvalidTokens := etiMark;
  aTokens.TextHint := 'Add recipients';
  aTokens.OnValidateToken := aEvents.ValidateToken;
  // the auto-complete list opens in the Registered and Trial editions
  aTokens.AutoCompleteItems.Add('anna@example.com');
  aTokens.AutoCompleteItems.Add('ben@example.com');
  aTokens.AutoCompleteItems.Add('sales@example.com');
  aTokens.AutoCompleteMatch := emmContains;
  aTokens.AutoCompleteMinChars := 2;
  aTokens.AddToken('team@example.com');
end;

function InvalidCount(aTokens: TsgcVCLTokenEdit): Integer;
var
  i: Integer;
begin
  Result := 0;
  for i := 0 to aTokens.Tokens.Count - 1 do
    if aTokens.TokenModel.IsTokenInvalid(i) then
      Inc(Result);
end;
```

Véase también

- Entrada de texto e IME
- Teclado y accesibilidad
- Ediciones

Valoración

Clases: TsgcVCLRating (unidad sgcEditors_VCL_Value) y TsgcFMXRating (unidad sgcEditors_FMX_Value)
La valoración es una fila de glifos dibujada a medida, que recibe el foco: estrellas, corazones, círculos o un glifo propio. Un clic elige el valor bajo el ratón, y pasar el ratón muestra una vista previa de un valor sin cambiarlo. Precision redondea el valor a glifos enteros o a mitades, o lo mantiene exacto para una media de solo lectura como 3.7, dibujada con un glifo parcialmente relleno.
La clase VCL desciende de TsgcVCLCustomEditor y la clase FireMonkey de TsgcFMXCustomEditor. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Value	Double	El valor, de 0 a Count, redondeado por Precision.
Count	Integer	El número de glifos, de 1 a 20. Por defecto 5.
Precision	TsgcEditorRatingPrecision	erpWhole (por defecto) redondea a glifos enteros, erpHalf a mitades, y erpExact no redondea. Asigne Count y Precision antes que Value.
Glyph	TsgcEditorRatingGlyph	ergStar (por defecto), ergHeart, ergCircle o ergCustom, que dibuja mediante OnDrawGlyph.
GlyphSize	Double	El tamaño de un glifo en píxeles lógicos. 0 (por defecto) es el mayor tamaño que cabe en el control.
Spacing	Double	Los píxeles lógicos entre dos glifos. Por defecto 4.
FillColor	TsgcUIColor	El color de los glifos rellenos. 0 (por defecto) es el ámbar de la paleta del tema.
EmptyColor	TsgcUIColor	El color de los glifos vacíos. 0 (por defecto) es el color de borde del tema.
AllowClear	Boolean	Un clic en el valor actual lo borra a 0, y Home va a 0. Por defecto False, y entonces el menor valor que puede elegir el usuario es un paso.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.
ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
RatingModel	TsgcRatingModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Valor cambiado.
OnHover	TsgcRatingHoverEvent	(Sender; aValue: Double). El valor bajo el ratón mientras muestra la vista previa, y -1 cuando termina la vista previa.
OnDrawGlyph	TsgcRatingDrawGlyphEvent	(Sender; aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aFill: Double). Dibuja un glifo de ergCustom en aRect, en píxeles de dispositivo. Se llama con aFill 0 para el glifo vacío y, después, otra vez con la parte rellena, de 0 a 1 desde la izquierda.

Teclado

Right y Up suman un paso de la precisión, Left y Down restan uno. Home va al menor valor, End a Count, y las teclas de dígitos asignan un valor entero.

Ejemplo: una nota sobre diez con un glifo propio

```
uses
  SysUtils, Classes, Controls, StdCtrls,
  sgcUICore_Canvas, sgcUICore_Theme, sgcEditors_Types, sgcEditors_VCL_Value;

type
  TScoreEvents = class
  public
    Status: TLabel;
    procedure DrawGlyph(Sender: TObject; aCanvas: TsgcUICanvas;
      const aRect: TsgcUIRect; aFill: Double);
    procedure Hover(Sender: TObject; aValue: Double);
  end;

procedure TScoreEvents.DrawGlyph(Sender: TObject; aCanvas: TsgcUICanvas;
  const aRect: TsgcUIRect; aFill: Double);
var
  vRight: Double;
  vColor: TsgcUIColor;
begin
  // called with aFill 0 for the empty glyph (content layer), then with the
  // fill, 0..1, for the filled part (overlay); aRect is in device pixels
  vRight := aRect.Right;
  vColor := $FFD0D5DD;
  if aFill > 0 then
  begin
    vRight := aRect.Left + (aRect.Right - aRect.Left) * aFill;
```

```
    vColor := $FF2E7D32;
  end;
  aCanvas.BeginPath;
  aCanvas.RoundRect(aRect.Left, aRect.Top, vRight, aRect.Bottom, 3, 3);
  aCanvas.FillPath(vColor);
end;

procedure TScoreEvents.Hover(Sender: TObject; aValue: Double);
begin
  // the value under the mouse, -1 when the preview ends
  if aValue < 0 then
    Status.Caption := ''
  else
    Status.Caption := FormatFloat('0.#', aValue) + ' of 10';
end;

procedure SetupScore(aRating: TsgcVCLRating; aEvents: TScoreEvents);
begin
  // Count and Precision before Value
  aRating.Count := 10;
  aRating.Precision := erpHalf;
  aRating.Value := 7.5;
  aRating.Glyph := ergCustom;
  aRating.GlyphSize := 14;
  aRating.Spacing := 2;
  aRating.AllowClear := True;
  aRating.OnDrawGlyph := aEvents.DrawGlyph;
  aRating.OnHover := aEvents.Hover;
end;
```

Véase también

- Valoración DB
- Control deslizante de rango
- Temas

Spin edit

Clases: TsgcVCLSpinEdit (unidad sgcEditors_VCL_Value) y TsgcFMXSpinEdit (unidad sgcEditors_FMX_Value)
El spin edit es un edit nativo para un número, con botones de subir y bajar. El valor es un Double: esvInteger lo redondea a números enteros y esvFloat a Decimals. Units es un sufijo atenuado, como kg o %, dibujado después del texto y que nunca forma parte de él. Un botón mantenido pulsado se repite, más rápido cuanto más tiempo se mantiene. La interpretación es tolerante: se ignoran los espacios y el separador de millares, y tanto . como , se aceptan como separador decimal.
La clase VCL desciende de TsgcVCLCustomEditor y la clase FireMonkey de TsgcFMXCustomEditor. Ambas publican las propiedades y los eventos de la tabla siguiente, además de los habituales de su framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit y los eventos de teclado y de ratón. Consulte Controles.

Propiedades

Propiedad	Tipo	Descripción
Value	Double	El valor. Asigne primero ValueType, MinValue y MaxValue, o Value se mantiene dentro de los límites antiguos.
MinValue, MaxValue	Double	Los límites. Valores iguales, como el 0 y 0 por defecto, significan sin límite.
Increment	Double	El paso de los botones, de Up, Down y de la rueda. Por defecto 1.
LargeIncrement	Double	El paso de PageUp y PageDown. Por defecto 10.
ValueType	TsgcEditorSpinValueType	esvInteger (por defecto) o esvFloat.
Decimals	Integer	Los decimales de un valor esvFloat. Por defecto 2.
DisplayFormat	string	Una máscara FormatFloat para el texto, como #,##0.00. Vacío (por defecto) escribe el valor con Decimals.
Units	string	Un sufijo atenuado después del texto, que no forma parte del valor.
Buttons	TsgcEditorSpinButtons	esbRight (por defecto) apila subir y bajar a la derecha, esbSides pone menos a la izquierda y más a la derecha, esbNone no tiene botones.
Wrap	Boolean	Un paso más allá de un límite primero se detiene en el límite, y el siguiente paso va al otro extremo. Por defecto False.
AllowEmpty	Boolean	Permite al usuario borrar el texto, lo que vacía el valor. Por defecto False.
IsEmpty	Boolean	Public, solo lectura. True mientras el valor está vacío.
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El color de acento ARGB de la selección, del relleno y del anillo de foco. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto. Un Color 0 sigue el tema.

ReadOnly	Boolean	El valor se muestra, pero el usuario no puede cambiarlo. Por defecto False.
AccessibleName	string	El nombre que anuncia un lector de pantalla antes del valor. Consulte Teclado y accesibilidad.
FormatSettings	TFormatSettings	Public. Los nombres, formatos y separadores que usa el control. Empieza con la configuración regional del usuario. Consulte Localización.
SpinModel	TsgcSpinModel	Public, solo lectura. El modelo del control. Model devuelve el mismo objeto como un TsgcEditorModel.

Métodos

Método	Devuelve	Descripción
Clear		Vacía el valor cuando AllowEmpty es True. En caso contrario, no hace nada.
SaveToPNG(aStream o aFileName, aScale = 1)		Dibuja el control con su tamaño actual en un PNG. aScale multiplica el tamaño. Consulte Salida sin ventana.
SaveToSVG(aStream o aFileName, aScale = 1)		Lo mismo en un SVG.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	Un valor nuevo: confirmado, avanzado por pasos o asignado en código.
OnParseError	TsgcSpinParseErrorEvent	(Sender; const aText: string; var aValue: Double; var aAccept: Boolean). El texto confirmado no es un número. Asigne un valor a aValue y True a aAccept para tomarlo. En caso contrario vuelve el último valor.
OnValidate	TsgcSpinValidateEvent	(Sender; var aValue: Double; var aAccept: Boolean). Se llama antes de asignar un valor nuevo. aAccept llega como True. Cambie aValue para ajustarlo, o asigne False a aAccept para conservar el valor anterior.

Teclado y ratón

Up y Down avanzan un Increment, PageUp y PageDown un LargeIncrement, y la rueda un Increment por muesca. Cada uno de ellos confirma antes el texto escrito. Enter y la pérdida de foco confirman el texto, y Escape lo restaura. Un botón mantenido pulsado se repite: un Increment por repetición, luego 2 desde la repetición 10, 5 desde la 20 y 10 desde la 30.

Ejemplo: un ángulo en medios grados

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Value;

type
```

```
  TAngleEvents = class
  public
    procedure Validate(Sender: TObject; var aValue: Double;
      var aAccept: Boolean);
  end;

procedure TAngleEvents.Validate(Sender: TObject; var aValue: Double;
  var aAccept: Boolean);
begin
  // aAccept comes in True; a typed angle snaps to half degrees
  aValue := Round(aValue * 2) / 2;
end;

procedure SetupAngle(aSpin: TsgcVCLSpinEdit; aEvents: TAngleEvents);
begin
  // the type and the limits before Value
  aSpin.ValueType := esvFloat;
  aSpin.Decimals := 1;
  aSpin.MinValue := 0;
  aSpin.MaxValue := 359.5;
  aSpin.Increment := 0.5;
  aSpin.LargeIncrement := 15;
  // a step past a limit stops on it, the next step goes to the other end
  aSpin.Wrap := True;
  aSpin.Units := 'deg';
  aSpin.Buttons := esbSides;
  aSpin.Value := 90;
  aSpin.OnValidate := aEvents.Validate;
end;
```

Véase también

- Spin edit DB
- Control deslizante de rango
- Localización

Selector de fecha DB

Clases: TsgcVCLDBDatePicker (unidad sgcEditors_VCL_DB) y TsgcFMXDBDatePicker (unidad sgcEditors_FMX_DB)
El selector de fecha DB enlaza la fecha de un selector de fecha con un campo de fecha o de fecha y hora. Desciende de TsgcVCLCustomDatePicker y TsgcFMXCustomDatePicker y publica las propiedades del selector de fecha salvo Date: el valor es el campo. Las demás propiedades, métodos y eventos publicados son los del selector de fecha.
Los controles data-aware forman parte de las ediciones Registered y Trial. Sus unidades están vacías en Community. Se enlazan mediante un TsgcEditorFieldLink, que solo usa la unidad DB, de modo que los controles FireMonkey no necesitan LiveBindings.

Propiedades

Propiedad	Tipo	Descripción
DataSource	TDataSource	El origen de datos del dataset.
DataField	string	El nombre del campo al que está enlazado el valor.
AllowEmpty	Boolean	Por defecto True, de modo que un campo nulo muestra un texto vacío y borrar el texto escribe nulo.
DisplayFormat, MinDate, MaxDate, DisabledDates, DisabledWeekDays, FirstDayOfWeek, ShowWeekNumbers, TextHint		Como en el selector de fecha.
ReadOnly	Boolean	True mantiene el control en solo lectura. También es de solo lectura mientras el campo no se puede modificar: sin dataset, un dataset o campo de solo lectura, o un campo calculado. Por defecto False.
Field	TField	Public, solo lectura. El campo enlazado, nil mientras el enlace no está activo.

Métodos

Método	Devuelve	Descripción
SaveToPNG, SaveToSVG		Como en el selector de fecha. También se aplican los demás métodos del control base.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El valor cambió, por el usuario, en código o porque el registro se movió.

OnDropDown, OnCloseUp, OnParseError, OnIsDateDisabled		Como en el selector de fecha.

Cómo se edita el campo

- El valor sigue al registro. Un campo nulo muestra un texto vacío.
- Un cambio del usuario pone el dataset en modo de edición y marca el valor como modificado. Asignar el valor en código también es un cambio. Cuando el dataset no puede entrar en modo de edición, vuelve el valor del campo.
- Enter y la pérdida de foco escriben el campo, y el dataset escribe un valor pendiente antes de hacer el post. Un campo que rechaza el valor lanza una excepción y el foco se queda en el control, como en TDBEdit.
- Escape restaura el valor del campo.
- El selector de fecha conserva la parte de hora de un TDateTimeField y cambia solo su fecha.

Ejemplo: una fecha de nacimiento

```
uses
  SysUtils, Classes, Controls, DB,
  sgcEditors_VCL_DB;

procedure BindBirthDate(aPicker: TsgcVCLDBDatePicker; aSource: TDataSource);
begin
  aPicker.DataSource := aSource;
  aPicker.DataField := 'BirthDate';
  aPicker.DisplayFormat := 'dd/mm/yyyy';
  aPicker.MaxDate := Date;
  // a null field shows an empty text with this hint
  aPicker.TextHint := 'Unknown';
end;

function BirthDateKnown(aPicker: TsgcVCLDBDatePicker): Boolean;
begin
  // Field is nil while the link is not active
  Result := (aPicker.Field <> nil) and not aPicker.Field.IsNull;
end;
```

Véase también

- Selector de fecha
- Selector de hora DB
- Inicio rápido data-aware
- TsgcEditorFieldLink

Selector de hora DB

Clases: TsgcVCLDBTimePicker (unidad sgcEditors_VCL_DB) y TsgcFMXDBTimePicker (unidad sgcEditors_FMX_DB)
El selector de hora DB enlaza la hora de un selector de hora con un campo de hora o de fecha y hora. Desciende de TsgcVCLCustomDatePicker y TsgcFMXCustomDatePicker y publica las propiedades del selector de hora salvo Time: el valor es el campo. Las demás propiedades, métodos y eventos publicados son los del selector de hora.
Los controles data-aware forman parte de las ediciones Registered y Trial. Sus unidades están vacías en Community. Se enlazan mediante un TsgcEditorFieldLink, que solo usa la unidad DB, de modo que los controles FireMonkey no necesitan LiveBindings.

Propiedades

Propiedad	Tipo	Descripción
DataSource	TDataSource	El origen de datos del dataset.
DataField	string	El nombre del campo al que está enlazado el valor.
AllowEmpty	Boolean	Por defecto True, de modo que un campo nulo muestra un texto vacío y borrar el texto escribe nulo.
DisplayFormat, MinTime, MaxTime, ListInterval, TextHint		Como en el selector de hora.
ReadOnly	Boolean	True mantiene el control en solo lectura. También es de solo lectura mientras el campo no se puede modificar: sin dataset, un dataset o campo de solo lectura, o un campo calculado. Por defecto False.
Field	TField	Public, solo lectura. El campo enlazado, nil mientras el enlace no está activo.

Métodos

Método	Devuelve	Descripción
SaveToPNG, SaveToSVG		Como en el selector de hora. También se aplican los demás métodos del control base.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El valor cambió, por el usuario, en código o porque el registro se movió.
OnDropDown, OnCloseUp, OnParseError		Como en el selector de hora.

Cómo se edita el campo

- El valor sigue al registro. Un campo nulo muestra un texto vacío.
- Un cambio del usuario pone el dataset en modo de edición y marca el valor como modificado. Asignar el valor en código también es un cambio. Cuando el dataset no puede entrar en modo de edición, vuelve el valor del campo.
- Enter y la pérdida de foco escriben el campo, y el dataset escribe un valor pendiente antes de hacer el post. Un campo que rechaza el valor lanza una excepción y el foco se queda en el control, como en TDBEdit.
- Escape restaura el valor del campo.
- El selector de hora conserva la parte de fecha de un TDateTimeField y cambia solo su hora.

Ejemplo: la hora de inicio de una reserva

```
uses
  SysUtils, Classes, Controls, DB,
  sgcEditors_VCL_DB;

procedure BindStartTime(aPicker: TsgcVCLDBTimePicker; aSource: TDataSource);
begin
  // a TDateTimeField keeps its date part, the picker edits the time
  aPicker.DataSource := aSource;
  aPicker.DataField := 'StartsAt';
  aPicker.DisplayFormat := 'hh:nn';
  aPicker.ListInterval := 15;
  aPicker.MinTime := EncodeTime(7, 0, 0, 0);
  aPicker.MaxTime := EncodeTime(22, 0, 0, 0);
end;
```

Véase también

- Selector de hora
- Selector de fecha DB
- Inicio rápido data-aware
- TsgcEditorFieldLink

Spin edit DB

Clases: TsgcVCLDBSpinEdit (unidad sgcEditors_VCL_DB) y TsgcFMXDBSpinEdit (unidad sgcEditors_FMX_DB)
El spin edit DB enlaza el valor de un spin edit con un campo numérico. Desciende de TsgcVCLSpinEdit y TsgcFMXSpinEdit, de modo que conserva los límites, los pasos, las unidades y los botones. Un campo entero lo cambia a esvInteger. Las demás propiedades, métodos y eventos publicados son los del spin edit.
Los controles data-aware forman parte de las ediciones Registered y Trial. Sus unidades están vacías en Community. Se enlazan mediante un TsgcEditorFieldLink, que solo usa la unidad DB, de modo que los controles FireMonkey no necesitan LiveBindings.

Propiedades

Propiedad	Tipo	Descripción
DataSource	TDataSource	El origen de datos del dataset.
DataField	string	El nombre del campo al que está enlazado el valor.
AllowEmpty	Boolean	Por defecto True, de modo que un campo nulo muestra un texto vacío y borrar el texto escribe nulo.
Value	Double	El valor del campo. No se guarda en el formulario.
ReadOnly	Boolean	True mantiene el control en solo lectura. También es de solo lectura mientras el campo no se puede modificar: sin dataset, un dataset o campo de solo lectura, o un campo calculado. Por defecto False.
Field	TField	Public, solo lectura. El campo enlazado, nil mientras el enlace no está activo.

Métodos

Método	Devuelve	Descripción
SaveToPNG, SaveToSVG		Como en el spin edit. También se aplican los demás métodos del control base.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El valor cambió, por el usuario, en código o porque el registro se movió.
OnParseError, OnValidate		Como en el spin edit. OnValidate se ejecuta antes de que el valor pase al campo.

Cómo se edita el campo

- El valor sigue al registro. Un campo nulo muestra un texto vacío.

- Un cambio del usuario pone el dataset en modo de edición y marca el valor como modificado. Asignar el valor en código también es un cambio. Cuando el dataset no puede entrar en modo de edición, vuelve el valor del campo.
- Enter y la pérdida de foco escriben el campo, y el dataset escribe un valor pendiente antes de hacer el post. Un campo que rechaza el valor lanza una excepción y el foco se queda en el control, como en TDBEdit.
- Escape restaura el valor del campo.
- Un campo entero cambia ValueType a esvInteger.

Ejemplo: una cantidad

```
uses
  SysUtils, Classes, Controls, DB,
  sgcEditors_VCL_DB;

procedure BindQuantity(aSpin: TsgcVCLDBSpinEdit; aSource: TDataSource);
begin
  // an integer field switches the spin edit to esvInteger
  aSpin.MinValue := 1;
  aSpin.MaxValue := 999;
  aSpin.LargeIncrement := 12;
  aSpin.Units := 'pcs';
  aSpin.DataSource := aSource;
  aSpin.DataField := 'Quantity';
  // AllowEmpty is True by default, so a null field shows an empty text
  aSpin.AllowEmpty := True;
end;
```

Ejemplo: un descuento en FireMonkey

```
uses
  System.SysUtils, System.Classes, Data.DB,
  sgcEditors_FMX_DB;

procedure BindDiscount(aSpin: TsgcFMXDBSpinEdit; aSource: TDataSource);
begin
  // the FireMonkey twin binds without LiveBindings
  aSpin.DataSource := aSource;
  aSpin.DataField := 'Discount';
  aSpin.MinValue := 0;
  aSpin.MaxValue := 100;
  aSpin.Units := '%';
end;
```

Véase también

- Spin edit
- Valoración DB
- Inicio rápido data-aware
- TsgcEditorFieldLink

Valoración DB

Clases: TsgcVCLDBRating (unidad sgcEditors_VCL_DB) y TsgcFMXDBRating (unidad sgcEditors_FMX_DB)
La valoración DB enlaza el valor de una valoración con un campo numérico. Desciende de TsgcVCLRating y TsgcFMXRating, de modo que conserva Count, Precision, los glifos y los colores. Con Precision erpExact y ReadOnly True muestra una media, como 3.7, de un campo calculado o agregado. Las demás propiedades, métodos y eventos publicados son los de la valoración.
Los controles data-aware forman parte de las ediciones Registered y Trial. Sus unidades están vacías en Community. Se enlazan mediante un TsgcEditorFieldLink, que solo usa la unidad DB, de modo que los controles FireMonkey no necesitan LiveBindings.

Propiedades

Propiedad	Tipo	Descripción
DataSource	TDataSource	El origen de datos del dataset.
DataField	string	El nombre del campo al que está enlazado el valor.
Value	Double	El valor del campo. No se guarda en el formulario.
ReadOnly	Boolean	True mantiene el control en solo lectura. También es de solo lectura mientras el campo no se puede modificar: sin dataset, un dataset o campo de solo lectura, o un campo calculado. Por defecto False.
Field	TField	Public, solo lectura. El campo enlazado, nil mientras el enlace no está activo.

Métodos

Método	Devuelve	Descripción
SaveToPNG, SaveToSVG		Como en la valoración. También se aplican los demás métodos del control base.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El valor cambió, por el usuario, en código o porque el registro se movió.
OnHover, OnDrawGlyph		Como en la valoración.

Cómo se edita el campo

- El valor sigue al registro. Un campo nulo no rellena ningún glifo.
- Un cambio del usuario pone el dataset en modo de edición y marca el valor como modificado. Asignar el valor en código también es un cambio. Cuando el dataset no puede entrar en modo de edición, vuelve el valor del campo.

- Enter y la pérdida de foco escriben el campo, y el dataset escribe un valor pendiente antes de hacer el post. Un campo que rechaza el valor lanza una excepción y el foco se queda en el control, como en TDBEdit.
- Escape restaura el valor del campo.
- Con AllowClear, un clic en el valor actual escribe 0 en el campo.

Ejemplo: estrellas y una media

```
uses
  SysUtils, Classes, Controls, DB,
  sgcEditors_Types, sgcEditors_VCL_DB;

procedure BindStars(aRating: TsgcVCLDBRating; aSource: TDataSource);
begin
  aRating.Count := 5;
  aRating.Precision := erpHalf;
  // a click on the current value writes 0 to the field
  aRating.AllowClear := True;
  aRating.DataSource := aSource;
  aRating.DataField := 'Stars';
end;

procedure ShowAverage(aRating: TsgcVCLDBRating; aSource: TDataSource);
begin
  // a read-only average such as 3.7: erpExact is not rounded
  aRating.Precision := erpExact;
  aRating.ReadOnly := True;
  aRating.DataSource := aSource;
  aRating.DataField := 'AvgStars';
end;
```

Véase también

- Valoración
- Spin edit DB
- Inicio rápido data-aware
- TsgcEditorFieldLink

Control deslizante de rango DB

Clases: TsgcVCLDBRangeSlider (unidad sgcEditors_VCL_DB) y TsgcFMXDBRangeSlider (unidad sgcEditors_FMX_DB)
El control deslizante de rango DB enlaza Value de un control deslizante de rango con DataField y, en RangeMode, EndValue con EndDataField, de modo que un control edita un rango guardado en dos campos. Desciende de TsgcVCLRangeSlider y TsgcFMXRangeSlider. Las demás propiedades, métodos y eventos publicados son los del control deslizante de rango.
Los controles data-aware forman parte de las ediciones Registered y Trial. Sus unidades están vacías en Community. Se enlazan mediante un TsgcEditorFieldLink, que solo usa la unidad DB, de modo que los controles FireMonkey no necesitan LiveBindings.

Propiedades

Propiedad	Tipo	Descripción
DataSource	TDataSource	El origen de datos del dataset.
DataField	string	El nombre del campo al que está enlazado el valor.
EndDataField	string	El campo de EndValue en RangeMode.
Value, EndValue	Double	Los valores de los dos campos. No se guardan en el formulario.
EndField	TField	Public, solo lectura. El campo de EndValue, nil mientras el enlace no está activo.
ReadOnly	Boolean	True mantiene el control en solo lectura. También es de solo lectura mientras el campo no se puede modificar: sin dataset, un dataset o campo de solo lectura, o un campo calculado. Por defecto False.
Field	TField	Public, solo lectura. El campo enlazado, nil mientras el enlace no está activo.

Métodos

Método	Devuelve	Descripción
SaveToPNG, SaveToSVG		Como en el control deslizante de rango. También se aplican los demás métodos del control base.

Eventos

Evento	Tipo	Descripción
OnChange	TNotifyEvent	El valor cambió, por el usuario, en código o porque el registro se movió.
OnChangeComplete, OnFormatLabel		Como en el control deslizante de rango.

Cómo se edita el campo

- El valor sigue al registro. Un campo nulo pone el tirador en Min, y un campo final nulo pone el tirador superior en Max.
- Un cambio del usuario pone el dataset en modo de edición y marca el valor como modificado. Asignar el valor en código también es un cambio. Cuando el dataset no puede entrar en modo de edición, vuelve el valor del campo.
- Enter y la pérdida de foco escriben el campo, y el dataset escribe un valor pendiente antes de hacer el post. Un campo que rechaza el valor lanza una excepción y el foco se queda en el control, como en TDBEdit.
- Escape restaura el valor del campo.
- En RangeMode el control es de solo lectura mientras el campo final no se puede modificar.

Ejemplo: un rango de precios en dos campos

```
uses
  SysUtils, Classes, Controls, DB,
  sgcEditors_VCL_DB;

procedure BindPriceRange(aSlider: TsgcVCLDBRangeSlider; aSource: TDataSource);
begin
  aSlider.Min := 0;
  aSlider.Max := 1000;
  aSlider.Step := 25;
  aSlider.ShowLabels := True;
  aSlider.TickInterval := 250;
  // Value goes to DataField and, in range mode, EndValue to EndDataField
  aSlider.RangeMode := True;
  aSlider.DataSource := aSource;
  aSlider.DataField := 'PriceFrom';
  aSlider.EndDataField := 'PriceTo';
end;

function RangeEditable(aSlider: TsgcVCLDBRangeSlider): Boolean;
begin
  // both fields must be writable for the range to change
  Result := (aSlider.Field <> nil) and aSlider.Field.CanModify and
    (aSlider.EndField <> nil) and aSlider.EndField.CanModify;
end;
```

Véase también

- Control deslizante de rango
- Valoración DB
- Inicio rápido data-aware
- TsgcEditorFieldLink

TsgcEditorModel

Unidad: sgcEditors_Types
TsgcEditorModel es la clase base de todos los modelos de sgcEditors. Es un TPersistent que solo usa la RTL y sgcUICore, de modo que funciona con VCL, con FireMonkey y sin ninguna ventana, en Windows y en Linux64. Un modelo guarda el valor y las opciones, recibe las teclas y el ratón, y se dibuja en dos capas: Render dibuja la capa de contenido y RenderOverlay el hover, el anillo de foco y las partes arrastradas. Cada control VCL y FireMonkey aloja un modelo, public como Model y como una propiedad tipada.

Modelo	Unidad	Usado por	Propiedad del control
TsgcCalendarModel	sgcEditors_Calendar	El calendario mensual y la lista desplegable de los selectores de fecha.	CalendarModel, DropDownCalendar
TsgcDatePickerModel	sgcEditors_Picker	Los selectores de fecha, hora y fecha y hora.	PickerModel
TsgcDateTimeDropDownModel	sgcEditors_Picker	La lista desplegable del selector de fecha y hora: un calendario y una lista de horas una al lado de la otra.	
TsgcEditorListModel	sgcEditors_List	La lista de horas y la lista de autocompletado del token edit.	DropDownTimeList, TokenModel.AutoCompleteList
TsgcSliderModel	sgcEditors_Slider	El control deslizante de rango.	SliderModel
TsgcColorPickerModel	sgcEditors_Color	El selector de color.	ColorModel
TsgcTokenModel	sgcEditors_Token	El token edit.	TokenModel
TsgcRatingModel	sgcEditors_Rating	La valoración.	RatingModel
TsgcSpinModel	sgcEditors_Spin	El spin edit.	SpinModel

Un modelo publica las mismas propiedades de valor que su control, como Date, Value o Tokens, además de miembros para los hosts, como HitTest y los rectángulos de sus partes.

Propiedades

Propiedad	Tipo	Descripción
Theme	TsgcUITheme	utLight (por defecto) o utDark. Consulte Temas.
AccentColor	TsgcUIColor	El acento ARGB. 0 (por defecto) usa el acento del tema.
TextStyle	TsgcUITextStyle	Size, Bold y Color del texto.

Enabled	Boolean	Un modelo deshabilitado no recibe entrada y dibuja los colores de deshabilitado. Por defecto True.
ReadOnly	Boolean	El modelo no recibe entrada que cambie el valor. Por defecto False.
Focused	Boolean	Public. Dibuja el anillo de foco en el overlay. Los hosts lo asignan cuando el control gana y pierde el foco.
HotPart	Integer	Public. La parte bajo el ratón, 0 cuando no hay ninguna. Cada modelo numera sus propias partes.
Painter	TsgcEditors-Painter	Public, solo lectura. Los colores del tema y el dibujo de marco, anillo de foco, chevron y flecha de spin del último renderizado.

Métodos

Método	Devuelve	Descripción
BeginUpdate, EndUpdate		Agrupa varios cambios. EndUpdate del par exterior envía cada tipo de cambio pendiente una vez.
IsUpdating	Boolean	True entre BeginUpdate y EndUpdate.
KeyDown(var aKey: Word; aShift)	Boolean	Una tecla, como código de tecla virtual de Windows. Las constantes CS_EDITORS_KEY_* contienen los códigos, y FireMonkey usa los mismos valores. True cuando el modelo la gestionó.
KeyPress(var aKey: Char)	Boolean	Un carácter. True cuando el modelo lo gestionó.
MouseDown, MouseUp(aButton, aShift, aX, aY)	Boolean	Un botón del ratón, embLeft, embRight o embMiddle, en aX y aY en píxeles lógicos. True cuando el modelo lo gestionó.
MouseMove(aShift, aX, aY), MouseWheel(aShift, aWheelDelta, aX, aY), MouseLeave	Boolean	El puntero y la rueda, en píxeles lógicos. Una muesca de la rueda es 120.
Render(aCanvas, aWidth, aHeight, aScale = 1)		Dibuja la capa de contenido en cualquier TsgcUICanvas. aWidth y aHeight son píxeles de dispositivo.
RenderOverlay(aCanvas, aWidth, aHeight, aScale = 1)		Dibuja el overlay sobre la capa de contenido.
SaveToPNG(aStream or aFileName, aWidth, aHeight, aScale = 1)		Dibuja las dos capas en un PNG. aWidth y aHeight son píxeles lógicos y aScale los multiplica.
SaveToSVG(aStream or aFileName, aWidth, aHeight, aScale = 1)		Lo mismo en un SVG.
MeasureText(aText, aFont)	Double	El ancho de un texto en píxeles de dispositivo: OnMeasure cuando está asignado, si no la fuente incrustada en sgcUICore.
Assign(aSource)		Copia las opciones y el valor de otro modelo de la misma clase.

Eventos

Evento	Tipo	Descripción
OnModelChange	TsgcEditorModelChangeEvent	(Sender; aKind: TsgcEditorChangeKind). eckContent: dibujar de nuevo la capa de contenido. eckOverlay: dibujar solo el overlay. eckValue: el valor cambió. eckLayout: las partes se movieron, así que hay que colocar de nuevo el edit interno.
OnMeasure	TsgcEditorMeasureEvent	function (const aText: string; const aFont: TsgcUIFontStyle): Double. Mide un texto con la fuente del host, en píxeles de dispositivo.

Un control asigna OnModelChange y OnMeasure de su propio modelo. Asígnelos solo en un modelo que cree usted mismo.

Tamaño y entrada

Un modelo se distribuye y hace el hit test en píxeles lógicos, con el tamaño del último Render o el tamaño dado a SetViewSize (Width y Height en TsgcCalendarModel). Llame a Render una vez, o indique el tamaño, antes de enviar entrada de ratón a un modelo que aloja usted mismo.

Ejemplo: manejar un modelo de control deslizante sin ventana

```
uses
  SysUtils, Classes,
  sgcEditors_Types, sgcEditors_Slider;

type
  TModelLog = class
  public
    Lines: TStrings;
    procedure ModelChange(Sender: TObject; aKind: TsgcEditorChangeKind);
  end;

procedure TModelLog.ModelChange(Sender: TObject; aKind: TsgcEditorChangeKind);
const
  CKinds: array [TsgcEditorChangeKind] of string = ('content', 'overlay',
    'value', 'layout');
begin
  Lines.Add(CKinds[aKind]);
end;

procedure DriveSlider(aLog: TModelLog);
var
  oSlider: TsgcSliderModel;
  vKey: Word;
begin
  oSlider := TsgcSliderModel.Create;
  try
    oSlider.OnModelChange := aLog.ModelChange;
    // one notification per kind for the whole block
    oSlider.BeginUpdate;
    try
      oSlider.Max := 10;
      oSlider.Step := 1;
      oSlider.Value := 4;
      oSlider.ShowLabels := True;
    finally
```

```
      oSlider.EndUpdate;
    end;
    // the input a host sends: neutral key codes, logical pixels
    vKey := CS_EDITORS_KEY_RIGHT;
    if oSlider.KeyDown(vKey, []) then
      aLog.Lines.Add('Value ' + FloatToStr(oSlider.Value));
    // both layers into a PNG, 300 x 48 logical pixels at twice the size
    oSlider.Focused := True;
    oSlider.SaveToPNG('slider.png', 300, 48, 2);
  finally
    oSlider.Free;
  end;
end;
```

Véase también

- Arquitectura
- Salida sin ventana
- TsgcEditorLocale

TsgcEditorLocale

Unidad: sgcEditors_Date
TsgcEditorLocale contiene los nombres de días y meses, los formatos de fecha y hora, los separadores y el primer día de la semana que los modelos de fecha usan para dibujar y para interpretar. Un control mantiene una configuración regional propia y la rellena desde su FormatSettings, así que en un formulario se asigna FormatSettings, consulte Localización. Use TsgcEditorLocale con un modelo que cree usted mismo, como un TsgcCalendarModel sin ventana, y con las funciones de fecha de la unidad.
TsgcCalendarModel y TsgcDatePickerModel tienen una propiedad Locale. El modelo no es propietario de la configuración regional, y nil (por defecto) usa sgcEditorDefaultLocale, una configuración regional compartida capturada de la del usuario en el primer uso. Rellene primero la configuración regional y asígnela después, porque el modelo se vuelve a dibujar cuando se asigna Locale, no cuando cambia una propiedad de la configuración regional.

Propiedades

Propiedad	Tipo	Descripción
ShortDayNames[aDay], LongDayNames[aDay]	string	Los nombres de un TsgcEditorWeekDay, de ewdMonday a ewdSunday.
ShortMonthNames[aMonth], LongMonthNames[aMonth]	string	Los nombres de un mes, de 1 a 12.
ShortDateFormat, LongDateFormat	string	Los formatos de fecha. Un DisplayFormat vacío de un selector de fecha usa ShortDateFormat.
ShortTimeFormat, LongTimeFormat	string	Los formatos de hora. Un DisplayFormat vacío de un selector de hora usa ShortTimeFormat.
TimeAMString, TimePMString	string	Los textos AM y PM de un formato de 12 horas.
DateSeparator, TimeSeparator	Char	Se escriben donde un formato tiene / y :.
FirstDayOfWeek	TsgcEditorWeekDay	El primer día de la semana que representa efdLocale.

Métodos

Método	Devuelve	Descripción
Create		Crea la configuración regional y llama a Capture.
Capture		Toma los nombres, los formatos y los separadores de la RTL, y el primer día de la semana del sistema operativo, el lunes cuando no tiene ninguno.
CaptureSettings(aSettings: TFormatSettings)		Toma los nombres, los formatos y los separadores de aSettings, como TFormatSettings.Create('de-DE'). FirstDayOfWeek se conserva.
Assign(aSource)		Copia otra configuración regional.

Funciones

Función	Devuelve	Descripción
sgcEditorDefaultLocale	TsgcEditor-Locale	La configuración regional compartida que usan los modelos cuando su Locale es nil. No la libere.
sgcEditorFormatDateTime(aDateTime, aFor-mat, aLocale = nil)	string	Escribe una fecha y hora con un DisplayFor-mat y los nombres de una configuración re-gional.
sgcEditorParseDateTime(aText, aFormat, aReference, out aDateTime, aLocale = nil)	Boolean	La interpretación tolerante de los selectores. Los campos que faltan en el texto proceden de aReference.
sgcEditorFirstDayOf(aFirstDay, aLocale = nil)	TsgcEditor-WeekDay	Resuelve un TsgcEditorFirstDayOfWeek, de modo que efdLocale da el FirstDayOfWeek de la configuración regional.

Eventos

TsgcEditorLocale no tiene eventos.

Ejemplo: un calendario alemán en un servidor

```
uses
  SysUtils, Classes,
  sgcEditors_Types, sgcEditors_Date, sgcEditors_Calendar;

procedure RenderGermanCalendar(const aFileName: string);
var
  oLocale: TsgcEditorLocale;
  oCalendar: TsgcCalendarModel;
begin
  // a locale of its own; the shared default locale stays as it is
  oLocale := TsgcEditorLocale.Create;
  try
    oLocale.CaptureSettings(TFormatSettings.Create('de-DE'));
    // CaptureSettings keeps the first day of the week, so set it
    oLocale.FirstDayOfWeek := ewdMonday;
    oCalendar := TsgcCalendarModel.Create;
    try
      // fill the locale first, then assign it; the model does not own it
      oCalendar.Locale := oLocale;
      oCalendar.ShowWeekNumbers := True;
      oCalendar.DisplayMonth := EncodeDate(2026, 10, 1);
      oCalendar.SaveToPNG(aFileName, 280, 240);
    finally
      oCalendar.Free;
    end;
  finally
    oLocale.Free;
  end;
end;

function GermanDateText(const aDate: TDateTime): string;
var
  oLocale: TsgcEditorLocale;
begin
  oLocale := TsgcEditorLocale.Create;
```

```
  try
    oLocale.CaptureSettings(TFormatSettings.Create('de-DE'));
    // the long day and month names of the locale
    Result := sgcEditorFormatDateTime(aDate, 'dddd, d. mmmm yyyy', oLocale);
  finally
    oLocale.Free;
  end;
end;
```

Véase también

- Localización
- Calendario mensual
- TsgcEditorModel

TsgcEditorFieldLink

Unidad: sgcEditors_Data
TsgcEditorFieldLink es el TDataLink de los controles data-aware. Solo usa la unidad DB, de modo que los controles VCL y FireMonkey lo comparten; FireMonkey no tiene TFieldDataLink. Lleva la semántica de TFieldDataLink, así que también puede usarlo para enlazar un control propio con un campo. Forma parte de las ediciones Registered y Trial, y la unidad está vacía en Community.

Propiedades

Propiedad	Tipo	Descripción
DataSource	TDataSource	Heredado de TDataLink. El origen de datos del dataset.
FieldName	string	El nombre del campo. Se resuelve con FieldByName cuando el enlace se activa y cuando cambian los campos del dataset.
Field	TField	Solo lectura. El campo, nil mientras el enlace no está activo o FieldName está vacío.
ReadOnly	Boolean	Heredado de TDataLink. True impide que el enlace edite.
CanModify	Boolean	Solo lectura. El enlace no es ReadOnly, el campo existe y TField.CanModify es True: el campo no es de solo lectura ni calculado y el dataset puede modificar.
Editing	Boolean	Solo lectura. True mientras el dataset está en estado de edición y el campo se puede modificar.
IsModified	Boolean	Solo lectura. True después de Modified, hasta que UpdateRecord escribe el campo, OnDataChange lo recarga o Editing cambia.

Métodos

Método	Devuelve	Descripción
Edit	Boolean	Pone el dataset en modo de edición a través del origen de datos, cuando el campo se puede modificar. True cuando después el enlace está editando.
Modified		Marca el valor del control como cambiado.
Reset		Recarga el control desde el campo mediante OnDataChange y borra IsModified.
UpdateRecord		Heredado de TDataLink. Dispara OnUpdateData cuando IsModified, de modo que el control escribe el campo una vez. El dataset lo llama antes de un post.

Eventos

Evento	Tipo	Descripción

OnDataChange	TNotifyEvent	El registro o el campo cambió: un desplazamiento, una cancelación, un post, una actualización o Reset. Muestre el valor del campo. No se dispara mientras Edit pone el dataset en modo de edición, porque entonces el control guarda el cambio pendiente del usuario.
OnUpdateData	TNotifyEvent	Escriba el valor del control en Field.
OnActiveChange	TNotifyEvent	El dataset se abrió o se cerró.
OnEditingChange	TNotifyEvent	Editing cambió. IsModified se borra.

Ejemplo: enlazar un control propio

```
uses
  SysUtils, Classes, Controls, DB,
  sgcEditors_Data, sgcEditors_VCL_Value;

type
  // binds a plain TsgcVCLRating to a numeric field, as the DB controls do
  TRatingBinder = class
  private
    FLink: TsgcEditorFieldLink;
    FRating: TsgcVCLRating;
    FLoading: Boolean;
    procedure LinkDataChange(Sender: TObject);
    procedure LinkUpdateData(Sender: TObject);
    procedure RatingChange(Sender: TObject);
    procedure RatingExit(Sender: TObject);
  public
    constructor Create(aRating: TsgcVCLRating; aSource: TDataSource;
      const aFieldName: string);
    destructor Destroy; override;
  end;

constructor TRatingBinder.Create(aRating: TsgcVCLRating; aSource: TDataSource;
  const aFieldName: string);
begin
  inherited Create;
  FRating := aRating;
  FRating.OnChange := RatingChange;
  FRating.OnExit := RatingExit;
  FLink := TsgcEditorFieldLink.Create;
  FLink.OnDataChange := LinkDataChange;
  FLink.OnUpdateData := LinkUpdateData;
  FLink.FieldName := aFieldName;
  FLink.DataSource := aSource;
end;

destructor TRatingBinder.Destroy;
begin
  FLink.Free;
  inherited;
end;

procedure TRatingBinder.LinkDataChange(Sender: TObject);
begin
  // the record moved or the field changed: show the value of the field
  FLoading := True;
  try
```

```
    if (FLink.Field = nil) or FLink.Field.IsNull then
      FRating.Value := 0
    else
      FRating.Value := FLink.Field.AsFloat;
  finally
    FLoading := False;
  end;
end;

procedure TRatingBinder.RatingChange(Sender: TObject);
begin
  // a change of the user: edit mode, then mark the value as modified;
  // when the dataset cannot be edited the field value returns
  if FLoading then
    Exit;
  if FLink.Edit then
    FLink.Modified
  else
    FLink.Reset;
end;

procedure TRatingBinder.LinkUpdateData(Sender: TObject);
begin
  // UpdateRecord and the post of the dataset call it when Modified
  FLink.Field.AsFloat := FRating.Value;
end;

procedure TRatingBinder.RatingExit(Sender: TObject);
begin
  FLink.UpdateRecord;
end;
```

Véase también

- Inicio rápido data-aware
- Valoración DB
- TsgcEditorModel

Entrada de texto e IME

Cada campo de texto de sgcEditors es un control de edición real: el texto de fecha u hora de los selectores, el número del spin edit, el campo hexadecimal del selector de color y la entrada después del último chip del token edit. El calendario mensual, el control deslizante de rango y la valoración no tienen texto y son controles dibujados que reciben el foco por sí mismos.

Por qué un edit nativo

El edit interno es un TEdit hijo sin borde en VCL y un TEdit de FMX en FireMonkey, colocado donde el modelo dibuja su texto. Como es el edit de la plataforma, la composición de IME para entrada en japonés, chino y coreano, el portapapeles, deshacer, el menú contextual y los lectores de pantalla funcionan sin código especial. El marco, los botones, las unidades y los chips que lo rodean los dibuja el modelo. En VCL el edit nunca es una parada de tabulación propia: el control le reenvía el foco, y el anillo de foco se dibuja mientras el edit tiene el foco.
En Windows, el canvas raster de cada control carga una vez los archivos de fuente de la plataforma y el edit interno usa la misma tipografía y el mismo tamaño en píxeles, de modo que el texto escrito y el texto dibujado coinciden. Las demás plataformas y la salida sin ventana usan Roboto, la fuente incrustada en sgcUICore.

Escribir una fecha o una hora

- El texto sigue DisplayFormat, formado por los segmentos d, dd, ddd, dddd, m, mm, mmm, mmmm, yy, yyyy, h, hh, n, nn, s, ss, am/pm y ampm más literales. Un `/` escribe el separador de fecha de la configuración regional y un `:` el separador de hora. Un DisplayFormat vacío usa la fecha corta, la hora corta o ambas, a partir del FormatSettings del control.
- La interpretación es tolerante. Cualquiera de `/ - . : ,` o un espacio separa los campos, se entienden los nombres de meses y días de la configuración regional, los años de dos dígitos tienen pivote en 50, y un año de cuatro dígitos al principio se lee como año, mes, día. Un campo que falta en el texto toma el año de hoy y la hora como 0, y un formato sin campos de hora conserva la hora del valor actual.
- Los selectores interpretan el texto en cada cambio y mueven con él el calendario abierto. El valor se confirma con Enter, al perder el foco y cuando se cierra la lista desplegable.
- Up y Down cambian el segmento bajo el cursor y lo vuelven a seleccionar, de modo que puede hacer girar el mes o la hora. Los campos dan la vuelta como el selector de fecha y hora de Windows, y después MinDate, MaxDate, MinTime y MaxTime acotan el resultado.
- Un texto que no se puede interpretar dispara OnParseError con el texto, el valor y aAccept en False. Asigne aValue y aAccept para tomar el texto, o déjelo y vuelve el último valor válido. Una fecha fuera de los límites o deshabilitada en el calendario se rechaza al confirmar.
- Con AllowEmpty en True, un texto vacío borra el valor e IsEmpty es True. TextHint se muestra mientras el edit está vacío: desde Delphi 2009 en VCL, y como el TextPrompt del edit en FireMonkey.

Escribir un número

El spin edit también interpreta con tolerancia. Se ignoran los espacios y el separador de millares de la configuración regional, y tanto `.` como `,` se aceptan como separador decimal. El texto se formatea con DisplayFormat, una máscara FormatFloat, o con Decimals cuando DisplayFormat está vacío. Units es un sufijo atenuado dibujado después del texto, que nunca forma parte de él. OnParseError funciona como en los selectores, y OnValidate permite cambiar o rechazar cada valor nuevo, con aAccept llegando como True.

Escribir tokens

- Cada carácter de Delimiters, una coma y un punto y coma por defecto, termina un token, y Enter siempre lo hace. Tab y la pérdida de foco también confirman el texto escrito. Pegar `a, b, c` crea tres chips.
- OnValidateToken recibe el texto recortado. Cámbielo para normalizar el token, o ponga aValid a False para rechazarlo. Con InvalidTokens en etiReject el token no se añade, y con etiMark se añade y se dibuja como no válido.
- AllowDuplicates, CaseSensitive y MaxTokens limitan lo que se añade. AddToken, RemoveToken y Clear cambian los tokens en código, y OnTokenAdded y OnTokenRemoved informan de cada cambio.

Ejemplo

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Date, sgcEditors_VCL_Value,
  sgcEditors_VCL_Token;

type
  TEntryRules = class(TComponent)
  public
    procedure DateParseError(Sender: TObject; const aText: string;
      var aValue: TDateTime; var aAccept: Boolean);
    procedure ValidateQuantity(Sender: TObject; var aValue: Double;
      var aAccept: Boolean);
    procedure ValidateEmail(Sender: TObject; var aText: string;
      var aValid: Boolean);
  end;

procedure TEntryRules.DateParseError(Sender: TObject; const aText: string;
  var aValue: TDateTime; var aAccept: Boolean);
begin
  // words the lenient parser does not know; aAccept comes in False
  if SameText(Trim(aText), 'today') then
  begin
    aValue := Date;
    aAccept := True;
  end
  else if SameText(Trim(aText), 'tomorrow') then
  begin
    aValue := Date + 1;
    aAccept := True;
  end;
end;

procedure TEntryRules.ValidateQuantity(Sender: TObject; var aValue: Double;
  var aAccept: Boolean);
begin
  // boxes of six: round the typed value up, aAccept comes in True
  aValue := Trunc((aValue + 5) / 6) * 6;
end;

procedure TEntryRules.ValidateEmail(Sender: TObject; var aText: string;
  var aValid: Boolean);
begin
  // aText is trimmed; normalise it, then accept or refuse it
  aText := LowerCase(aText);
  aValid := (Pos('@', aText) > 1) and (Pos('.', aText) > Pos('@', aText));
end;

procedure SetupEntry(aRules: TEntryRules; aPicker: TsgcVCLDatePicker;
  aSpin: TsgcVCLSpinEdit; aTokens: TsgcVCLTokenEdit);
begin
  aPicker.DisplayFormat := 'ddd d mmm yyyy';
  aPicker.AllowEmpty := True;
  aPicker.TextHint := 'Delivery date';
  aPicker.OnParseError := aRules.DateParseError;

  aSpin.ValueType := esvInteger;
  aSpin.Increment := 6;
  aSpin.OnValidate := aRules.ValidateQuantity;

  // a space also ends a token; refused addresses stay as marked chips
  aTokens.Delimiters := ',; ';
  aTokens.InvalidTokens := etiMark;
  aTokens.OnValidateToken := aRules.ValidateEmail;
end;
```

Véase también

- Teclado y accesibilidad
- Localización
- Selector de fecha
- Token edit

Teclado y accesibilidad

Ninguna función de sgcEditors necesita un ratón. Los controles dibujados reciben el foco por sí mismos, y los controles con texto reenvían el foco a su edit interno. Un control con foco dibuja un anillo de foco de 2 píxeles en el color de acento, en su capa de overlay.

Teclas

Control	Teclas
Calendario mensual	Las flechas se mueven un día y una semana y omiten los días deshabilitados. PageUp y PageDown se mueven un mes, con Ctrl un año. Enter o Espacio selecciona. Shift con las flechas extiende un rango. Ctrl+Up reduce a la vista de año y de década, y Enter vuelve a ampliar.
Selectores	F4 o Alt+Down abre y cierra la lista desplegable. Up y Down cambian el segmento bajo el cursor, o se mueven por la lista desplegable abierta. Enter confirma, Escape cierra la lista desplegable y restaura el valor.
Spin edit	Up y Down avanzan por Increment, PageUp y PageDown por LargeIncrement. La rueda del ratón avanza por Increment, y un botón mantenido pulsado se repite más rápido cuanto más tiempo se mantiene.
Control deslizante de rango	Las flechas avanzan por Step, PageUp y PageDown por LargeStep, Home y End van a Min y Max. En modo de rango, véanse los modelos de foco más abajo.
Valoración	Las flechas cambian el valor en un paso de Precision, Home y End van al menor valor y a Count, y un dígito asigna ese valor. Con AllowClear, 0 o elegir de nuevo el valor actual lo borra.
Selector de color	Las flechas se mueven entre las muestras o mueven el cursor del cuadrado y de las barras, Enter o Espacio selecciona, Home y End van a los extremos. El edit hexadecimal acepta textos #RGB, #RGBA, #RRGGBB y #RRGGBBAA.
Token edit	Enter o un delimitador añade el token escrito. Retroceso en una entrada vacía da el foco al último chip, y un segundo Retroceso lo quita. Left y Right recorren los chips, Delete quita el chip con foco, Escape vuelve a la entrada. Up y Down se mueven por la lista de autocompletado.

Modelos de foco

La mayoría de los controles son una sola parada de tabulación. Dos controles tienen dos partes, y VCL y FireMonkey llegan a ellas de forma distinta, porque FireMonkey mueve el foco con Tab antes de que el control vea la tecla.

Control	VCL	FireMonkey
Control deslizante de rango en modo de rango	Tab va del tirador inferior al superior, el siguiente Tab sale del control. Shift+Tab llega al tirador superior. Ctrl+Left o Ctrl+Down selecciona el tirador inferior, Ctrl+Right o Ctrl+Up el superior.	Una sola parada de tabulación. El foco empieza en el tirador inferior, y un clic conserva el tirador que se agarró. Ctrl+Left o Ctrl+Down selecciona el tirador inferior, Ctrl+Right o Ctrl+Up el superior.
Selector de color	Dos paradas de tabulación: las partes y después el edit hexadecimal. Enter en las partes selecciona el color y no llega al botón predeterminado del formulario. Un selector de solo lectura es una sola parada de tabulación.	Una sola parada de tabulación con dos paradas internas. Ctrl con las flechas pasa de las partes al edit hexadecimal y viceversa.

En FireMonkey el foco lo pone en cola el framework, y OnEnter y OnExit se disparan una vez por control incluso cuando el foco se mueve entre el control y su edit interno.

Accesibilidad

- Los edits internos son edits nativos, de modo que los lectores de pantalla leen y editan su texto como de costumbre.
- Los controles VCL dibujados responden a WM_GETTEXT con `AccessibleName: value` mediante el proxy de ventana de MSAA, y cada cambio del valor llama a NotifyWinEvent con EVENT_OBJECT_VALUECHANGE, de modo que un lector de pantalla anuncia una nueva posición del control deslizante o una nueva valoración.
- Los controles FireMonkey ponen el mismo texto en su Hint, respetando un Hint propio.
- El texto de los dos temas integrados tiene un contraste de al menos 4.5:1, y los colores derivados de un acento personalizado se ajustan para mantenerlo. QA lo comprueba.

Ejemplo

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_VCL_Date, sgcEditors_VCL_Value, sgcEditors_VCL_Color;

procedure SetupAccess(aCalendar: TsgcVCLMonthCalendar;
  aSlider: TsgcVCLRangeSlider; aRating: TsgcVCLRating;
  aColor: TsgcVCLColorPicker);
begin
  // screen readers announce "AccessibleName: value"
  aCalendar.AccessibleName := 'Departure date';
  aSlider.AccessibleName := 'Price range';
  aRating.AccessibleName := 'Hotel rating';
  aColor.AccessibleName := 'Label colour';

  // the tab order: calendar, slider (two stops in range mode), rating,
  // colour picker (the swatches, then the hex edit)
  aCalendar.TabOrder := 0;
  aSlider.TabOrder := 1;
  aRating.TabOrder := 2;
  aColor.TabOrder := 3;

  aSlider.RangeMode := True;
  aSlider.Step := 5;
  aSlider.LargeStep := 50;
  aRating.AllowClear := True;
end;
```

Véase también

- Entrada de texto e IME
- Temas
- Control deslizante de rango
- Selector de color

Temas

Cada control tiene un Theme, un AccentColor y un TextStyle. Los colores del marco, del hover, del estado pulsado, de la selección y del estado deshabilitado se derivan de ellos mediante TsgcEditorsPainter, de modo que una propiedad cambia el aspecto de todo el control.

Propiedad	Efecto
Theme	utLight (por defecto) o utDark, los conjuntos de colores claro y oscuro de sgcUICore. utCustom usa el conjunto claro.
AccentColor	El color de la selección, de los tiradores, del relleno y del anillo de foco, como un TsgcUIColor ARGB, por ejemplo $FF2A78D6. 0 usa el acento del tema, y un color sin alfa se toma como opaco.
TextStyle	Size (11 por defecto), Bold y Color del texto. Un Color 0 sigue el color de texto del tema.
Enabled	Un control deshabilitado mezcla su superficie, texto, borde y acento con el fondo.

Colores derivados

Los rellenos de hover y de pulsado son mezclas claras del acento sobre la superficie. El texto sobre un relleno de acento, como el día seleccionado, es un color oscuro o claro, el que se lea mejor, y el relleno se oscurece o se aclara hasta que ese texto alcanza un contraste de 4.5:1. La banda de selección de un rango mantiene el mismo contraste con el texto normal. Así, cualquier color de acento sigue siendo legible en los dos temas.

Colores de los controles

- La valoración tiene FillColor y EmptyColor. Un FillColor 0 es ámbar y un EmptyColor 0 sigue el tema. Glyph elige ergStar, ergHeart, ergCircle o ergCustom, que dibuja mediante OnDrawGlyph.
- El selector de color muestra Swatches, por defecto 24 colores de las paletas de sgcUICore y grises.

El ancho de un chip de token depende del tema y de la fuente, de modo que un cambio de Theme o de TextStyle mide de nuevo cada chip y distribuye de nuevo el control.

Ejemplo

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcUICore_Theme, sgcEditors_Types, sgcEditors_VCL, sgcEditors_VCL_Value;

procedure ApplyEditorsTheme(aForm: TForm; aDark: Boolean);
var
  i: Integer;
  oEditor: TsgcVCLCustomEditor;
begin
  // every sgcEditors control descends from TsgcVCLCustomEditor
  for i := 0 to aForm.ComponentCount - 1 do
    if aForm.Components[i] is TsgcVCLCustomEditor then
    begin
      oEditor := TsgcVCLCustomEditor(aForm.Components[i]);
      if aDark then
        oEditor.Theme := utDark
      else
        oEditor.Theme := utLight;
      // ARGB; 0 returns to the accent of the theme
      oEditor.AccentColor := $FF8E44AD;
```

```
      oEditor.TextStyle.Size := 12;
    end;
end;

procedure StyleRating(aRating: TsgcVCLRating);
begin
  aRating.Glyph := ergHeart;
  aRating.FillColor := $FFE0245E;
  aRating.EmptyColor := 0;
  aRating.Precision := erpHalf;
end;
```

Véase también

- Teclado y accesibilidad
- Salida sin ventana

Localización

Cada control tiene una propiedad pública FormatSettings. Empieza con la configuración regional del usuario, y asignarla cambia los nombres, los formatos y los separadores que usa el control. Cada control conserva su propia copia, de modo que dos controles de un mismo formulario pueden usar dos configuraciones regionales.

Fechas y horas

- El calendario y los selectores capturan de FormatSettings los nombres cortos y largos de los días, los nombres cortos y largos de los meses, los formatos cortos y largos de fecha y hora, las cadenas AM y PM y los separadores de fecha y hora.
- FirstDayOfWeek es efdLocale por defecto, que toma el primer día de la semana del sistema operativo, el lunes cuando el sistema no tiene ninguno. efdMonday, efdSaturday y efdSunday lo fuerzan.
- WeekNumbering es ewnISO8601 por defecto: la semana 1 es la que contiene el primer jueves del año. ewnSimple cuenta la semana 1 como la que contiene el 1 de enero, empezando en el primer día de la semana.
- DisplayFormat vacío significa la fecha corta de la configuración regional para el selector de fecha, la hora corta para el selector de hora, y ambas, separadas por un espacio, para el selector de fecha y hora. La interpretación entiende los nombres de meses y días de la configuración regional. Consulte Entrada de texto e IME.

Números

El spin edit y las etiquetas y la sugerencia de valor del control deslizante de rango dan formato a sus números con los separadores decimal y de millares de FormatSettings. DisplayFormat del spin edit y LabelFormat del control deslizante son máscaras FormatFloat, y OnFormatLabel del control deslizante puede escribir cualquier texto, como una moneda o una unidad.

Texto de los controles

El título del calendario, los nombres de los días y la lista de horas proceden de la configuración regional, y las unidades, las sugerencias y los nombres accesibles son propiedades suyas.

Ejemplo

Un calendario alemán con números de semana ISO, un selector de fecha y hora con los formatos cortos alemanes, y un spin edit y un control deslizante con separadores alemanes:

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Date, sgcEditors_VCL_Value;

procedure UseGerman(aCalendar: TsgcVCLMonthCalendar;
  aPicker: TsgcVCLDateTimePicker; aSpin: TsgcVCLSpinEdit;
  aSlider: TsgcVCLRangeSlider);
var
  vSettings: TFormatSettings;
begin
  vSettings := TFormatSettings.Create('de-DE');

  // day and month names, formats and separators come from FormatSettings
  aCalendar.FormatSettings := vSettings;
  aCalendar.FirstDayOfWeek := efdMonday;
  aCalendar.ShowWeekNumbers := True;
  aCalendar.WeekNumbering := ewnISO8601;

  // empty DisplayFormat: the short date and the short time of the locale
```

```
  aPicker.FormatSettings := vSettings;
  aPicker.DisplayFormat := '';

  // 1.234,50 EUR
  aSpin.FormatSettings := vSettings;
  aSpin.ValueType := esvFloat;
  aSpin.DisplayFormat := '#,##0.00';
  aSpin.Units := 'EUR';

  aSlider.FormatSettings := vSettings;
  aSlider.LabelFormat := '#,##0';
end;
```

Véase también

- Entrada de texto e IME
- DPI
- TsgcEditorLocale

DPI

sgcEditors distribuye cada control en píxeles lógicos, píxeles a 96 DPI, y lo dibuja a la escala de la pantalla. No hace falta hacer nada en su código.

VCL

- La escala de un control es su PPI dividido entre 96. El contenido y el overlay se rasterizan a esa escala, y las posiciones del ratón se dividen entre ella antes de llegar al modelo.
- Cuando el formulario pasa a un monitor con otro DPI, ChangeScale y WM_DPICHANGED mueven y escalan el edit interno y su fuente junto con el control.
- La lista desplegable se dibuja al PPI del monitor en el que se abre, de modo que se ve nítida cuando el formulario ocupa dos monitores. Delphi 7 usa Screen.PixelsPerInch.

FireMonkey

El raster de un control es su Width multiplicado por la escala de la escena, de modo que se ve nítido en pantallas Retina y Android de DPI alto. El modelo conserva su disposición lógica.

Salida sin ventana

SaveToPNG y SaveToSVG reciben el tamaño en píxeles lógicos y una escala. Una escala de 2 escribe un PNG con el doble de ancho y de alto, la misma disposición que un control en una pantalla de 192 DPI.

Texto medido

Los anchos de texto, como el ancho de un chip de token, dependen de la escala. El token edit mantiene una caché de chips medidos, con clave en el medidor, la escala y el tema, de modo que un cambio de DPI mide de nuevo cada chip y distribuye de nuevo el control.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcEditors_Calendar;

procedure ExportCalendarIcons(const aFolder: string);
var
  oModel: TsgcCalendarModel;
begin
  oModel := TsgcCalendarModel.Create;
  try
    oModel.Date := EncodeDate(2026, 10, 14);
    oModel.ShowWeekNumbers := True;
    // the same 280 x 240 logical layout at 96, 144 and 192 DPI:
    // 280 x 240, 420 x 360 and 560 x 480 device pixels
    oModel.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'cal_100.png',
      280, 240, 1);
    oModel.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'cal_150.png',
      280, 240, 1.5);
    oModel.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'cal_200.png',
      280, 240, 2);
  finally
    oModel.Free;
```

```
  end;
end;
```

Véase también

- Salida sin ventana
- Temas

Salida sin ventana

Los modelos de sgcEditors se renderizan sin ventana. Úselos para dibujar un control en un informe, en un documento o en una página web, para hacer una prueba de instantánea, o para renderizar en un servidor, en Windows o en Linux64.

Modelos

Cree un modelo, como TsgcCalendarModel, TsgcDatePickerModel, TsgcSliderModel, TsgcColorPickerModel, TsgcTokenModel, TsgcRatingModel o TsgcSpinModel, asigne sus propiedades y llame a uno de estos métodos de TsgcEditorModel. Dibujan la capa de contenido y el overlay.

Método	Escribe
SaveToPNG(aStream, aWidth, aHeight, aScale)	Un PNG en un stream.
SaveToPNG(aFileName, aWidth, aHeight, aScale)	Un archivo PNG.
SaveToSVG(aStream, aWidth, aHeight, aScale)	Un SVG en un stream.
SaveToSVG(aFileName, aWidth, aHeight, aScale)	Un archivo SVG.

aWidth y aHeight son píxeles lógicos, y aScale, 1 por defecto, los multiplica. La salida sin ventana usa Roboto, la fuente incrustada en sgcUICore, de modo que el mismo modelo da los mismos píxeles en cualquier máquina. Los métodos Render y RenderOverlay dibujan en cualquier TsgcUICanvas que usted posea.

Controles

Cada control VCL y FireMonkey también tiene SaveToPNG y SaveToSVG con un stream o un nombre de archivo y una escala. Renderizan el control con su tamaño actual, con el texto de su edit interno dibujado por el modelo.

Linux64

Desde Delphi 10.2, las unidades independientes del framework compilan para Linux64, de modo que un servicio Linux puede renderizar el mismo calendario, control deslizante o valoración que una aplicación de escritorio.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcUICore_Theme, sgcEditors_Types, sgcEditors_Calendar, sgcEditors_Spin,
  sgcEditors_Rating;

procedure RenderEditors(aStream: TStream);
var
  oCalendar: TsgcCalendarModel;
  oSpin: TsgcSpinModel;
  oRating: TsgcRatingModel;
begin
  // a range in a dark month calendar, as an SVG file
  oCalendar := TsgcCalendarModel.Create;
  try
    oCalendar.Theme := utDark;
    oCalendar.SelectionMode := esmRange;
    oCalendar.RangeStart := EncodeDate(2026, 10, 12);
```

```
    oCalendar.RangeEnd := EncodeDate(2026, 10, 16);
    oCalendar.DisplayMonth := EncodeDate(2026, 10, 1);
    oCalendar.SaveToSVG('calendar.svg', 280, 240);
  finally
    oCalendar.Free;
  end;

  // a spin edit with units, as a PNG at twice the size
  oSpin := TsgcSpinModel.Create;
  try
    oSpin.ValueType := esvFloat;
    oSpin.Decimals := 1;
    oSpin.Value := 21.5;
    oSpin.Units := 'C';
    oSpin.SaveToPNG('spin.png', 120, 28, 2);
  finally
    oSpin.Free;
  end;

  // three and a half stars, as a PNG into a stream
  oRating := TsgcRatingModel.Create;
  try
    oRating.Precision := erpHalf;
    oRating.Value := 3.5;
    oRating.SaveToPNG(aStream, 160, 32);
  finally
    oRating.Free;
  end;
end;
```

Véase también

- DPI
- Arquitectura
- TsgcEditorModel

Ediciones

sgcEditors se compila desde el mismo código fuente para todas las ediciones.

Edición	Incluye
Registered	Todo: los nueve controles para VCL y FireMonkey, los cinco controles data-aware de cada framework, el cuadrado HSV y la barra alfa del selector de color, la lista de autocompletado del token edit y los dos tiradores del control deslizante de rango.
Trial	Las mismas características que Registered. Una compilación hecha con SGC_TRIAL muestra un mensaje de que es una versión de prueba cuando se crea un componente después de que haya pasado la ventana de lanzamiento indicada en la compilación. No cambia nada más.
Community	Los nueve controles para VCL y FireMonkey sin las características avanzadas que se indican a continuación, y sin aviso de prueba.

Las características avanzadas

Las características avanzadas se compilan bajo el define SGC_EDITORS_ADVANCED, que sgcVer.inc desactiva en la edición Community. Esto es exactamente lo que cambia:

Característica	En Community
Selector de color	ecpHSV y ecpAlpha en Parts se ignoran: el cuadrado, la barra de tono y la barra alfa no se dibujan, el alfa es siempre $FF y el texto hexadecimal es #RRGGBB. Las muestras y el edit hexadecimal se mantienen.
Token edit	AutoCompleteItems, AutoCompleteMatch, AutoCompleteMinChars y AutoCompleteMaxItems se mantienen, de modo que un formulario guardado por una compilación Registered se carga, pero la lista nunca se abre.
Control deslizante de rango	RangeMode se mantiene, pero se dibuja y se edita un tirador. EndValue se conserva y no se muestra.
Controles data-aware	Las unidades sgcEditors_Data, sgcEditors_VCL_DB y sgcEditors_FMX_DB se compilan como unidades vacías, de modo que las diez clases data-aware y su registro no existen. La paleta muestra 18 componentes en lugar de 28.

Community conserva el calendario completo, los tres selectores con sus listas desplegables, el spin edit con unidades, la valoración de media estrella, la validación de tokens, los temas, el teclado, el DPI y la salida PNG y SVG sin ventana.

Código para todas las ediciones

La edición es una elección en tiempo de compilación en sgcVer.inc, y SGC_COMMUNITY también desactiva SGC_TRIAL. Una unidad que debe compilar con Community y usa un miembro que solo existe en la edición avanzada incluye sgcVer.inc y pone ese código dentro de {$IFDEF SGC_EDITORS_ADVANCED}. Un formulario con un control data-aware no se carga en una compilación Community, porque allí la clase no existe. sgcEditors_Const tiene la versión y la edición como constantes.

```
{$I sgcVer.inc}

uses
  SysUtils, Classes, Controls,
  sgcEditors_Const, sgcEditors_Types, sgcEditors_VCL_Color,
  sgcEditors_VCL_Token;

procedure SetupPickers(aColor: TsgcVCLColorPicker; aTags: TsgcVCLTokenEdit);
begin
  // the full picker: Community ignores ecpHSV and ecpAlpha
  aColor.Parts := [ecpSwatches, ecpHSV, ecpAlpha, ecpHex];
{$IFDEF SGC_EDITORS_ADVANCED}
  // the auto-complete list opens only in the Registered and Trial editions
  aTags.AutoCompleteItems.CommaText := 'Delphi,FireMonkey,VCL,Linux';
  aTags.AutoCompleteMatch := emmContains;
{$ENDIF}
end;

function EditionText: string;
begin
  // for example 'sgcEditors 2026.10.0 Registered'
  Result := CS_SGC_EDITORS_FULLNAME + ' ' + CS_SGC_EDITORS_EDITION;
end;
```

Véase también

- Instalación
- Inicio rápido data-aware

Descripción general de .NET

sgcEditors .NET es una versión gestionada en C# de la biblioteca sgcEditors de Delphi, no un wrapper alrededor de ella. Las clases C# viven en el espacio de nombres **esegece.sgcEditors** y conservan los nombres de Delphi, de modo que **TsgcCalendarModel**, **TsgcSpinModel**, **TsgcEditorLocale** y las enumeraciones se leen igual en los dos lenguajes, y los nueve controles siguen el patrón de nombres de Delphi en todos los hosts: **TsgcVCLDatePicker** es **TsgcWinFormsDatePicker**, **TsgcWpfDatePicker** y **TsgcAvaloniaDatePicker**. Los controles se dibujan con la versión .NET de **sgcUICore**, el mismo canvas, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe, alrededor de un cuadro de texto alojado del framework para el texto escrito.

Paquetes

La versión se divide en cinco paquetes NuGet, de modo que un proyecto solo incorpora el host que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcEditors	Los modelos de los nueve controles (**TsgcCalendarModel**, **TsgcDatePickerModel**, **TsgcSliderModel**, **TsgcRatingModel**, **TsgcSpinModel**, **TsgcColorPickerModel**, **TsgcTokenModel**), los modelos de lista y de lista desplegable, las matemáticas de fechas, la configuración regional y la salida SVG sin ventana. Sin dependencia nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcEditors.Rendering	La salida PNG sin ventana, los métodos de extensión **SaveToPNG** y **RenderToRaster** de **sgcEditors_Raster**, dibujados en el canvas SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcEditors.WinForms	Los nueve controles **TsgcWinForms** sobre **TsgcWinFormsCustomEditor**.	net472, net8.0-windows, net9.0-windows
esegece.sgcEditors.Wpf	Los nueve controles **TsgcWpf** sobre **TsgcWpfCustomEditor**.	net472, net8.0-windows, net9.0-windows
esegece.sgcEditors.Avalonia	Los nueve controles **TsgcAvalonia** sobre **TsgcAvaloniaCustomEditor**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0

Los tres paquetes de host dependen de esegece.sgcEditors.Rendering, que depende de esegece.sgcEditors, de modo que una sola línea `PackageReference` trae todo lo que necesita un host. Consulte Instalación desde NuGet.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcEditors, .Rendering, .WinForms, .Wpf y .Avalonia	Todo: los nueve controles de los tres hosts, los temas, el soporte de teclado y accesibilidad, la salida PNG y SVG sin ventana y las tres características avanzadas: el cuadrado HSV y la barra alfa del selector de color, la lista de autocompletado del token edit y el segundo tirador del deslizador de rango.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcEditors.WinForms.Community.	Todo excepto las tres características avanzadas. Las propiedades se mantienen, de modo que el mismo código compila en las dos ediciones: **Parts** ignora **ecpHSV** y **ecpAlpha** (el alfa es siempre $FF), **AutoCompleteItems** nunca abre una lista, y **RangeMode** dibuja y edita un tirador mientras se conserva **EndValue**.

Es la misma división que las ediciones de Delphi, consulte Ediciones. Los controles data-aware de Delphi no tienen equivalente en .NET en ninguna edición: las propiedades de valor se enlazan con el enlace de datos de cada framework, consulte Enlace de datos en .NET.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (controles TsgcVCL)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	VCL y FMX	Windows, .NET Framework 4.7.2 y .NET 8 y 9. Los controles se pueden declarar en XAML.
Avalonia	FMX (controles TsgcFMX)	Windows, Linux y macOS, .NET 8 y 9.
Sin ventana	los modelos y sus SaveToPNG y SaveToSVG	Cualquier plataforma de los frameworks de destino: SVG con esegece.sgcEditors, PNG con esegece.sgcEditors.Rendering.

Todos los hosts tienen la misma API: las propiedades, los métodos y los eventos de los controles de Delphi con los mismos nombres, más **Theme**, **AccentColor**, **TextStyle**, **ReadOnly**, **FormatSettings**, **DropDown**, **CloseUp**, **SaveToPNG** y **SaveToSVG** en la clase base. Consulte Clases de .NET para los nombres de las clases y Diferencias de .NET respecto a Delphi para lo que cambia respecto a Delphi.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Enlace de datos en .NET
- .NET Headless Rendering
- Demos de .NET
- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, los paquetes del núcleo y del renderizado llegan con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcEditors.WinForms
dotnet add package esegece.sgcEditors.Wpf
dotnet add package esegece.sgcEditors.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcEditors.WinForms.Community
dotnet add package esegece.sgcEditors.Wpf.Community
dotnet add package esegece.sgcEditors.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcEditors.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los Community, nunca los dos: contienen los mismos tipos en el mismo espacio de nombres.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:sgc="clr-namespace:esegece.sgcEditors;assembly=esegece.sgcEditors.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. El cuadro de texto interno toma su aspecto del tema Fluent. Asigne el espacio de nombres en XAML con `xmlns:sgc="using:esegece.sgcEditors"`.
- Sin ventana: esegece.sgcEditors solo basta para SVG. Añada esegece.sgcEditors.Rendering para PNG, trae SkiaSharp y sus recursos nativos para Windows y Linux.

En C# todo host necesita `using esegece.sgcEditors;`, y `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUITextStyle** y los demás tipos de los temas.

Véase también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos de .NET

Inicio rápido WinForms

Este inicio rápido construye en código un pequeño formulario de reserva con cuatro de los nueve controles WinForms: un selector de fecha con calendario desplegable, un spin edit con unidades, un deslizador de rango y un token edit.

Pasos

1. Cree un proyecto Windows Forms para `net8.0-windows` y añada el paquete **esegece.sgcEditors.WinForms** (**esegece.sgcEditors.WinForms.Community** en la edición Community).
2. Añada `using esegece.sgcEditors;` y `using esegece.sgcUICore;`.
3. Cree los controles, defina sus propiedades y añádalos al formulario. Son controles WinForms normales: Location, Size, Dock, Anchor, TabIndex y Enabled funcionan como siempre.
4. Asigne los eventos. **OnChange** lo dispara cada control cuando cambia su valor, los demás eventos son los del control de Delphi.

Ejemplo

Notas

- Los valores de fecha y hora son fechas OLE Automation en un `double`, el TDateTime de Delphi. Convierta con `DateTime.ToOADate` y `DateTime.FromOADate`.
- El deslizador de rango espera entre **BeginInit** y **EndInit** antes de limitar **Value** y **EndValue** a **Min** y **Max**, de modo que el orden de las asignaciones no importa. El diseñador de Windows Forms llama a los dos, mediante `ISupportInitialize`.
- **Theme** es por control. Cada control desciende de **TsgcWinFormsCustomEditor**, de modo que un bucle sobre los controles define el tema de un formulario entero.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Enlace de datos en .NET
- Selector de fecha

Inicio rápido WPF

Los controles WPF se pueden declarar en XAML como cualquier otro control. Este inicio rápido declara un selector de fecha y hora, una valoración de media estrella y un selector de color, y define el resto en el code behind.

Pasos

1. Cree un proyecto WPF para `net8.0-windows` y añada el paquete **esegece.sgcEditors.Wpf** (**esegece.sgcEditors.Wpf.Community** en la edición Community).
2. Asigne el espacio de nombres en la ventana: `xmlns:sgc="clr-namespace:esegece.sgcEditors;assembly=esegece.sgcEditors.Wpf"`.
3. Declare los controles. Las enumeraciones toman los nombres de sus valores, como `Precision="erpHalf"`. Un control sin Width y Height toma un tamaño por defecto (160 x 28 para la mayoría de los selectores, 200 x 28 para el selector de fecha y hora, 280 x 240 para el calendario mensual).
4. Asigne los eventos en el code behind.

Ejemplo

La ventana:

```
<Window x:Class="QuickStart.QuickStartWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:sgc="clr-namespace:esegece.sgcEditors;assembly=esegece.sgcEditors.Wpf"
        Title="Review" Width="360" Height="300">
  <StackPanel Margin="12">
    <sgc:TsgcWpfDateTimePicker x:Name="dttVisit" Width="220" Height="28" HorizontalAlignment="Left" />
    <sgc:TsgcWpfRating x:Name="rtgScore" Precision="erpHalf" Value="3.5" Margin="0,8,0,0"
                       Width="160" Height="32" HorizontalAlignment="Left" />
    <sgc:TsgcWpfColorPicker x:Name="clrTag" Margin="0,8,0,0" Width="240" Height="180"
                            HorizontalAlignment="Left" />
  </StackPanel>
</Window>
```

El code behind:

Notas

- Las propiedades de valor son propiedades de dependencia que enlazan en los dos sentidos por defecto: **Date**, **Time** y **DateTime** de los selectores, **Value** y **EndValue** del deslizador, **Value** de la valoración y del spin edit, **SelectedColor** del selector de color. Consulte Enlace de datos en .NET.
- El selector de color tiene **SelectedWpfColor**, el `System.Windows.Media.Color` opaco de **SelectedColor**.
- Las listas desplegables son popups de WPF colocados debajo del control, o encima de él cerca de la parte inferior de la pantalla, y dibujados con el DPI del monitor del control.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Enlace de datos en .NET
- Valoración

Inicio rápido Avalonia

Los controles Avalonia se ejecutan en Windows, Linux y macOS. Este inicio rápido es una aplicación Avalonia completa en código: un calendario mensual que selecciona un rango de hasta 14 días y un selector de hora con una lista desplegable cada 30 minutos.

Pasos

1. Cree un proyecto de consola `net8.0`, defina `<OutputType>WinExe</OutputType>` y añada **Avalonia.Desktop** y **Avalonia.Themes.Fluent** 11.3 y el paquete **esegece.sgcEditors.Avalonia** (**esegece.sgcEditors.Avalonia.Community** en la edición Community).
2. Añada el tema Fluent a la aplicación. El cuadro de texto interno de los selectores toma su aspecto de él.
3. Cree los controles y añádalos a un panel. En XAML asigne el espacio de nombres con `xmlns:sgc="using:esegece.sgcEditors"`.

Ejemplo

Notas

- Una hora es la fracción de un día de una fecha OLE Automation, por lo que `TimeSpan.TotalDays` es un valor de hora.
- Las propiedades de valor son propiedades con estilo que enlazan en los dos sentidos por defecto. Consulte Enlace de datos en .NET.
- La selección de rango forma parte de todas las ediciones. El segundo tirador del deslizador de rango es la característica avanzada, consulte Descripción general de .NET.
- **AutomationProperties.Name** da nombre a un control para los lectores de pantalla, su valor se expone mediante el patrón de valor de su automation peer.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Enlace de datos en .NET
- Calendario mensual

Clases de .NET

Cada clase de Delphi conserva su nombre en .NET. Los controles VCL y FireMonkey pasan a ser los controles WinForms, WPF y Avalonia con el prefijo del host, y las clases independientes del framework son las mismas en todos los hosts.

Clases de los hosts

Delphi	WinForms	WPF	Avalonia	Model (propiedad del host)
TsgcVCLMonthCalendar, TsgcFMXMonthCalendar	TsgcWinForms-MonthCalendar	TsgcWpfMonth-Calendar	TsgcAvalonia-MonthCalendar	TsgcCalendarModel (CalendarModel)
TsgcVCLDatePicker, TsgcFMXDatePicker	TsgcWinForms-DatePicker	TsgcWpfDate-Picker	TsgcAvalonia-DatePicker	TsgcDatePicker-Model (PickerModel)
TsgcVCLTimePicker, TsgcFMXTimePicker	TsgcWinForms-TimePicker	TsgcWpfTime-Picker	TsgcAvalonia-TimePicker	TsgcDatePicker-Model (PickerModel)
TsgcVCLDateTimePicker, TsgcFMXDateTimePicker	TsgcWinForms-DateTimePicker	TsgcWpfDate-TimePicker	TsgcAvalonia-DateTimePicker	TsgcDatePicker-Model (PickerModel)
TsgcVCLRangeSlider, TsgcFMXRangeSlider	TsgcWinForms-RangeSlider	TsgcWpfRangeSlider	TsgcAvalonia-RangeSlider	TsgcSliderModel (SliderModel)
TsgcVCLRating, TsgcFMX-Rating	TsgcWinForms-Rating	TsgcWpfRating	TsgcAvalonia-Rating	TsgcRatingModel (RatingModel)
TsgcVCLSpinEdit, TsgcFMXSpinEdit	TsgcWin-FormsSpinEdit	TsgcWpfSpinEdit	TsgcAvaloniaSpinEdit	TsgcSpinModel (SpinModel)
TsgcVCLColorPicker, TsgcFMXColorPicker	TsgcWinForms-ColorPicker	TsgcWpfColor-Picker	TsgcAvalonia-ColorPicker	TsgcColorPicker-Model (ColorModel)
TsgcVCLTokenEdit, TsgcFMXTokenEdit	TsgcWinForms-TokenEdit	TsgcWpfTokenEdit	TsgcAvalonia-TokenEdit	TsgcTokenModel (TokenModel)
TsgcVCLCustomEditor, TsgcFMXCustomEditor	TsgcWinForms-CustomEditor	TsgcWpfCustomEditor	TsgcAvalonia-CustomEditor	TsgcEditorModel (Model)
TsgcVCLCustomDatePicker, TsgcFMXCustomDate-Picker	TsgcWinForms-CustomDatePicker	TsgcWpfCustomDatePicker	TsgcAvalonia-CustomDate-Picker	TsgcDatePicker-Model (PickerModel)
el TEdit interno	TsgcWinFormsEditorInnerEdit (a TextBox)	TsgcWpfEditorInnerEdit (a TextBox)	TsgcAvaloniaEditorInnerEdit (a TextBox)	
la lista desplegable	TsgcWinFormsEditorPopup (a	TsgcWpfEditor-Popup (a Popup)	TsgcAvaloniaEditorPopup (a Popup)	TsgcDateTime-DropDownModel,

	ToolStripDrop-Down)			TsgcEditorListMo-del

Las clases base derivan de **TsgcUIWinFormsControl**, **TsgcUIWpfControl** y **TsgcUIAvaloniaControl** de sgcUI-Core. Las clases de accesibilidad son **TsgcWinFormsEditorAccessibleObject**, **TsgcWpfEditorAutomationPeer** y **TsgcAvaloniaEditorAutomationPeer**; informan del nombre y del valor del control.

Miembros de todos los controles

Miembro	Descripción
Model	El modelo del control, del tipo de la tabla anterior. La propiedad tipada (CalendarMo-del, SpinModel...) devuelve el mismo objeto.
InnerEdit	El cuadro de texto alojado, null en los controles sin texto escrito (calendario men-sual, deslizador de rango, valoración).
Popup	La lista desplegable de los selectores y de la lista de autocompletado de los tokens.
Theme, AccentColor, TextStyle	El aspecto: **TsgcUITheme** (utLight, utDark), el color de acento como `uint` ($AARRGGBB, 0 es el color del tema) y el tamaño y el peso de la fuente.
ReadOnly	El valor se puede leer y copiar, pero no cambiar.
FormatSettings	Una copia de **TsgcEditorFormatSettings** con la que el control da formato e inter-preta, la cultura actual por defecto. Cambie la copia y vuelva a asignarla.
DroppedDown, Drop-Down(), CloseUp(bool)	El estado de la lista desplegable: abrirla, cerrarla aceptando o sin aceptar.
OnChange	Se dispara cuando cambia el valor, Sender es el control.
SaveToPNG, Save-ToSVG	Dibujan el control tal como se ve ahora en un flujo o en un archivo, a una escala.

Tipos independientes del framework

Tipo	Descripción
TsgcEditorModel	El modelo base: tema, foco, habilitado, hover, teclado, ratón, **Render**, **RenderOver-lay** y **SaveToSVG**. Consulte TsgcEditorModel.
TsgcEditorLocale	Los nombres de los días y de los meses y el primer día de la semana, tomados de la configuración de formato. Consulte TsgcEditorLocale.
TsgcEditorFor-matSettings	El subconjunto del TFormatSettings de Delphi que leen los editores. `Create()` to-ma la cultura actual, `Create(CultureInfo)` otra.
TsgcEditorStrings	El TStrings de Delphi: **DisabledDates**, **Swatches**, **Tokens**, **AutoCompleteItems**.
TsgcEditorListModel	La lista de la lista desplegable de horas y del autocompletado de los tokens.
TsgcDateTimeDrop-DownModel	La lista desplegable del selector de fecha y hora: un calendario junto a una lista de horas.

sgcEditors_Types, sgcEditors_Date	Clases estáticas con las funciones libres y las constantes de Delphi: códigos de tecla (CS_EDITORS_KEY_*), formato de números, conversión hexadecimal y HSV, las matemáticas de fechas.
sgcEditors_Raster	En esegece.sgcEditors.Rendering: los métodos de extensión **SaveToPNG** y **RenderToRaster** de todos los modelos.

Enumeraciones

Las enumeraciones conservan los nombres y los prefijos de valor de Delphi: **TsgcEditorSelectionMode** (esm), **TsgcEditorFirstDayOfWeek** (efd), **TsgcEditorWeekNumbering** (ewn), **TsgcEditorCalendarView** (ecv), **TsgcEditorOrientation** (eor), **TsgcEditorTickPlacement** (etp), **TsgcEditorInvalidTokens** (eti), **TsgcEditorMatchMode** (emm), **TsgcEditorRatingPrecision** (erp), **TsgcEditorRatingGlyph** (erg), **TsgcEditorSpinValueType** (esv) y **TsgcEditorSpinButtons** (esb). Los conjuntos de Delphi son enumeraciones `[Flags]` combinadas con `|`: **TsgcEditorWeekDays**, **TsgcEditorColorParts**, **TsgcEditorChangeKinds** y **TsgcEditorShiftState**.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Controles
- Arquitectura

Diferencias de .NET respecto a Delphi

La versión .NET conserva los nombres, los valores por defecto, el comportamiento del teclado y las reglas de renderizado de Delphi. La salida SVG de los modelos es idéntica byte a byte a la de Delphi, y la salida PNG coincide con ella dentro de la tolerancia del rasterizador SkiaSharp. Las diferencias siguientes proceden de los frameworks de los hosts y de la biblioteca de clases de .NET.

Sin controles data-aware

En .NET no hay controles DB ni **TsgcEditorFieldLink**, en ninguna edición. Las propiedades de valor se enlazan en su lugar con el enlace de datos de cada framework: `Binding` de WinForms mediante `INotifyPropertyChanged`, propiedades de dependencia de WPF y propiedades con estilo de Avalonia, ambas en los dos sentidos por defecto. Consulte Enlace de datos en .NET.

Sin streaming de componentes

En .NET no hay DFM, por lo que no hay un editor de listas de cadenas en tiempo de diseño para **DisabledDates** y **Swatches**: rellénelas en código. El deslizador de rango aplaza el límite de **Value** y **EndValue** a entre **BeginInit** y **EndInit**, el gancho que llaman el diseñador de Windows Forms y los cargadores de XAML, donde Delphi espera a Loaded.

Valores, eventos y listas

- TDateTime es un `double`, una fecha OLE Automation: `DateTime.ToOADate` y `DateTime.FromOADate` la convierten. Una hora es la fracción de un día.
- TsgcUIColor es un `uint`, $AARRGGBB, donde 0 significa el color del tema. **SelectedVCLColor** es **SelectedWinFormsColor** (`System.Drawing.Color`), **SelectedWpfColor** (`System.Windows.Media.Color`) y **SelectedAvaloniaColor** (`Avalonia.Media.Color`), todos opacos.
- Los eventos son propiedades delegate con los nombres de Delphi (**OnChange**, **OnSelect**, **OnTokenAdded**...). Asigne un manejador con `=`. Un parámetro `var` de Delphi es un parámetro `ref`, y una lambda con un parámetro `ref` declara el tipo de todos los parámetros. **TNotifyEvent** es **TsgcUINotifyEvent**, y Sender es el control.
- Una propiedad TStrings es un **TsgcEditorStrings**, con **Add**, **Insert**, **Delete**, **Clear**, **IndexOf** (que ignora mayúsculas y minúsculas), **Assign**, **BeginUpdate** y **EndUpdate**. A los objetos de la lista de tokens se llega con **GetObject**.
- Una función de Delphi sin parámetros sigue siendo un método, como `IsRange()`, `CellCount()` o `HexText()`. Un conjunto de Delphi es una enumeración `[Flags]` combinada con `|`.
- TFormatSettings es **TsgcEditorFormatSettings**. La propiedad **FormatSettings** devuelve una copia: cámbiela y vuelva a asignarla, como con el record de Delphi.

Cuadro de texto, teclado y foco en cada host

- El texto escrito es un `TextBox` alojado del framework, por lo que el cursor, la selección, el portapapeles, deshacer y el IME son los del framework. **TextHint** es el cue banner en WinForms, una pista dibujada por el edit interno en WPF y el `Watermark` en Avalonia.
- Tab, Enter y Escape se comportan como en Delphi en los tres hosts. En Avalonia, una tecla que gestiona el control no llega a un manejador KeyDown del propio control.
- El modelo trabaja en píxeles lógicos. WPF y Avalonia maquetan en DIP, por lo que el edit interno y la lista desplegable mantienen el mismo tamaño con cualquier DPI; WinForms los escala con el DPI del monitor.
- Accesibilidad: WinForms informa del nombre y del valor mediante el `AccessibleObject`, WPF mediante un `AutomationPeer` con el patrón de valor, Avalonia mediante `AutomationProperties` y su automation peer. No hay texto de ventana como en la VCL.
- En WPF, **Tokens** del token edit es una propiedad CLR, no una propiedad de dependencia; **OnChange** informa de sus cambios. En Avalonia es una propiedad directa que se notifica en cada cambio.

Véase también

- Descripción general de .NET
- Clases de .NET
- Enlace de datos en .NET
- Teclado y accesibilidad

Enlace de datos en .NET

Los controles data-aware de Delphi enlazan un control con un campo de un TDataSet mediante **TsgcEditorFieldLink**. En .NET las propiedades de valor de los controles normales se enlazan con el enlace de datos de cada framework, en las dos ediciones, de modo que no hay controles DB aparte.

Control	Propiedades de valor
MonthCalendar	Date, RangeStart, RangeEnd
DatePicker, TimePicker, DateTimePicker	Date, Time, DateTime (las tres son la misma fecha OLE Automation vista como fecha, hora o ambas), IsEmpty (solo lectura)
RangeSlider	Value, EndValue
Rating	Value
SpinEdit	Value, IsEmpty (solo lectura)
ColorPicker	SelectedColor (`uint`), más el color de solo lectura del tipo del host
TokenEdit	Tokens

WinForms

Los controles WinForms implementan `INotifyPropertyChanged` y lo disparan para las propiedades de valor, de modo que un `Binding` con `DataSourceUpdateMode.OnPropertyChanged` escribe de vuelta al instante, mientras el usuario escribe, incrementa o arrastra.

WPF

Las propiedades de valor son propiedades de dependencia registradas para enlazar en los dos sentidos por defecto: **DateProperty**, **TimeProperty**, **DateTimeProperty**, **RangeStartProperty**, **RangeEndProperty**, **ValueProperty**, **EndValueProperty**, **SelectedColorProperty** y la **IsEmptyProperty** de solo lectura. El control puede limitar un valor enlazado a sus límites y entonces escribe de vuelta el valor limitado. **Tokens** es una propiedad CLR y no se enlaza.

Avalonia

Las propiedades de valor son propiedades con estilo que enlazan en los dos sentidos por defecto, con los mismos nombres que en WPF. **IsEmptyProperty**, **SelectedAvaloniaColorProperty** y **TokensProperty** son propiedades directas: enlace **Tokens** en los dos sentidos a un **TsgcEditorStrings**, el setter copia las cadenas.
La demo BindingWpf enlaza un selector de fecha, un selector de hora, un spin edit, una valoración y un deslizador de rango a una lista de reservas, consulte Demos de .NET.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Demos de .NET
- Inicio rápido data-aware

Renderizado sin ventana en .NET

Cada control dibuja a través de su modelo, y los modelos no necesitan ventana. Un servicio, una prueba o un informe puede crear un modelo, definir sus propiedades y guardarlo como SVG o PNG en cualquier plataforma, Linux incluido.

- **SaveToSVG** es un método de **TsgcEditorModel** en esegece.sgcEditors, sin dependencia nativa.
- **SaveToPNG** es un método de extensión de **sgcEditors_Raster** en esegece.sgcEditors.Rendering, dibujado con SkiaSharp. **RenderToRaster** devuelve el propio canvas raster; quien llama lo libera.

Los dos reciben el flujo o el nombre del archivo, el ancho y el alto en píxeles lógicos y una escala opcional: una escala de 2 escribe una imagen de 560 x 480 para un calendario de 280 x 240. Los dos dibujan la capa de contenido y después la capa de superposición (hover, anillo de foco, un tirador arrastrado), de modo que la salida es lo que muestra el control.

Ejemplo

Notas

- Los hosts también tienen **SaveToPNG** y **SaveToSVG**, con el tamaño del control.
- Los modelos dan formato a números y fechas con la cultura actual. Para obtener la misma salida en todas las máquinas, asigne `TsgcEditorFormatSettings.Create(CultureInfo.InvariantCulture)` a **FormatSettings** de los modelos spin y slider, o del control del host.
- Las listas desplegables también son modelos (**TsgcDateTimeDropDownModel**, **TsgcEditorListModel**), dibujados con los mismos métodos.

Véase también

- Descripción general de .NET
- Clases de .NET
- Salida sin ventana

Demos de .NET

Las demos están en `.net\demos\Editors`, una carpeta por demo con su proyecto, su `Program.cs` y un README.md. Construyen su interfaz de usuario en código, sin XAML. Cada demo tiene un modificador `--test` que ejecuta una autocomprobación fuera de pantalla, escribe líneas PASS y FAIL y termina con el número de comprobaciones fallidas, como el modificador `/test` de las demos de Delphi.

Demo	Host	Muestra
Gallery	WinForms	Los nueve controles en una ventana: un calendario mensual con números de semana, un selector de fecha, un selector de hora con una lista de 15 minutos, un selector de fecha y hora, un spin edit con unidades, una valoración, un deslizador de rango con marcas, un token edit con lista de autocompletado y un selector de color con muestras, el cuadrado HSV, alfa y un campo hexadecimal. Un conmutador de tema, un conmutador de solo lectura, exportación a PNG y un registro de eventos.
GalleryWpf	WPF	La misma ventana con los controles **TsgcWpf**.
GalleryAvalonia	Avalonia	La misma ventana con los controles **TsgcAvalonia**, para Windows, Linux y macOS.
BookingForm	WinForms	Un formulario de reserva de hotel con siete controles: un calendario en modo de rango para la estancia (como máximo 28 noches), selectores de hora de entrada y de salida con límites y una lista cada 30 minutos, un spin edit para los huéspedes, un token edit para hasta seis peticiones con lista de autocompletado, una valoración para la categoría del hotel y un selector de color con muestras y hexadecimal para el color de la etiqueta.
BindingWpf	WPF	MVVM: una lista de reservas en un view model, y un selector de fecha, un selector de hora, un spin edit, una valoración y un deslizador de rango enlazados a la reserva seleccionada, el equivalente en .NET de la demo data-aware de Delphi.

Las demos Gallery muestran el cuadrado HSV, la barra alfa, el segundo tirador del deslizador de rango y la lista de autocompletado solo en la edición Registered, y BookingForm muestra la lista de autocompletado solo en ella. En la edición Community el mismo código se ejecuta sin ellos.

Compilar las demos

Dentro del árbol del producto las demos se compilan contra los proyectos de la biblioteca. El zip de las demos usa por defecto los paquetes NuGet de la edición Community. Cada demo se compila y se ejecuta en las dos ediciones. Compile una demo con los paquetes Registered definiendo **SgcEditorsEdition**:

```
dotnet build Gallery\Gallery.csproj -p:SgcEditorsEdition=Registered
dotnet run --project BookingForm\BookingForm.csproj -- --test
```

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Enlace de datos en .NET

Solución de problemas

Problemas habituales de sgcEditors y sus causas.

Problema	Causa y solución
Una fecha escrita vuelve al valor anterior.	El texto no se pudo interpretar, o la fecha está fuera de MinDate y MaxDate, o está deshabilitada por DisabledDates, DisabledWeekDays u OnIsDateDisabled. Gestione OnParseError para aceptar sus propias palabras, y llame a IsDateEnabled antes de asignar una fecha en código.
El selector de hora no tiene botón de lista desplegable.	ListInterval es 0. Asígnele los minutos entre los elementos de la lista, 30 por defecto. Los elementos quedan entre MinTime y MaxTime.
Un valor asignado en código cambia.	El valor queda acotado por los límites. Asigne MinValue, MaxValue, ValueType y Decimals antes que Value. MinValue igual a MaxValue significa sin límite. El Max de un control deslizante con Step baja a la cuadrícula de pasos, como en un input range de HTML.
Las elecciones en una lista desplegable de FireMonkey se pierden.	Un clic fuera de la lista desplegable la cierra y deshace las elecciones hechas mientras estaba abierta. Ciérrela con Enter o con una elección.
Tab no pasa al segundo tirador en FireMonkey.	FireMonkey mueve el foco con Tab. Use Ctrl+Right o Ctrl+Up para el tirador superior y Ctrl+Left o Ctrl+Down para el tirador inferior.
El selector de color no tiene cuadrado ni barra alfa.	Parts no incluye ecpHSV ni ecpAlpha, o la compilación es Community, que los ignora.
La lista de autocompletado nunca se abre.	AutoCompleteItems está vacío, se han escrito menos de AutoCompleteMinChars caracteres, o la compilación es Community.
El control deslizante de rango muestra un tirador.	RangeMode es False, o la compilación es Community.
Class TsgcVCLDBDatePicker not found when a form loads.	La aplicación está compilada con la edición Community, que no tiene controles data-aware.
Un control data-aware no se puede editar.	ReadOnly es True, falta DataSource o su dataset, el dataset o el campo es de solo lectura, el campo es calculado o, para un control deslizante en modo de rango, el campo final no se puede modificar.
Asignar Value de un control data-aware pone el dataset en modo de edición.	Un valor asignado en código cuenta como un cambio del usuario. Para cambiar el registro sin editarlo, escriba el campo.
El PNG sin ventana usa otra fuente distinta de la de la pantalla.	La pantalla usa la fuente de la plataforma, Segoe UI en Windows, y la salida sin ventana usa el Roboto incrustado, para que la salida sea la misma en todas partes.
La entrada IME no funciona en el calendario, el control deslizante o la valoración.	Estos controles no tienen texto. IME funciona en los edits internos de los selectores, del spin edit, del selector de color y del token edit.

Un formulario con cientos de editores se abre despacio.	La mayor parte del tiempo se va en la creación de las ventanas de los controles y de sus edits internos: unos 340 ms para 270 controles. Cree los controles de las páginas ocultas cuando la página se muestre por primera vez.

Ejemplo

```
uses
  SysUtils, Classes, Controls,
  sgcEditors_Types, sgcEditors_VCL_Date, sgcEditors_VCL_Value;

procedure SafeSetDate(aPicker: TsgcVCLDatePicker; const aDate: TDateTime);
begin
  // a date outside MinDate..MaxDate or disabled is refused on commit
  if aPicker.IsDateEnabled(aDate) then
    aPicker.Date := aDate
  else
    aPicker.Clear;
end;

procedure SetupTimeList(aPicker: TsgcVCLTimePicker);
begin
  // ListInterval 0 removes the drop-down button of a time picker
  aPicker.ListInterval := 15;
  aPicker.MinTime := EncodeTime(8, 0, 0, 0);
  aPicker.MaxTime := EncodeTime(18, 0, 0, 0);
end;

procedure SetupPercent(aSpin: TsgcVCLSpinEdit);
begin
  // set the limits and the type before Value, else Value is clamped
  // to the old limits
  aSpin.ValueType := esvFloat;
  aSpin.Decimals := 1;
  aSpin.MinValue := 0;
  aSpin.MaxValue := 100;
  aSpin.Value := 37.5;
  aSpin.Units := '%';
end;
```

Véase también

- Entrada de texto e IME
- Inicio rápido data-aware
- Ediciones

Introducción

sgcGauges añade medidores de instrumento a las aplicaciones Delphi y C++Builder, para VCL, para FireMonkey y sin ningún framework de interfaz de usuario. Hay diez tipos de medidor: radial, lineal, digital, LED, barra de LEDs, anillo de progreso, VU meter, reloj, brújula y anunciador. Cada tipo existe como un control VCL con el prefijo TsgcVCL, como un control FireMonkey con el prefijo TsgcFMX y como una clase de modelo independiente del framework, como TsgcRadialGauge, que genera PNG y SVG sin ventana.
Un medidor tiene una o más escalas, cada una con sus rangos de color, uno o más punteros (aguja, marcador, barra o rango) y umbrales que lanzan un evento cuando un valor los cruza. Un nuevo valor mueve el puntero con una animación y una curva de suavizado. Los medidores tienen temas claro y oscuro, se adaptan a su tamaño y al DPI, y el Value de un medidor puede escribirse desde cualquier hilo.
Cada medidor se dibuja en dos capas. La capa estática contiene la esfera, las escalas, las marcas, las etiquetas y los rangos, y solo se vuelve a dibujar cuando cambia la configuración. La capa dinámica contiene los punteros y el texto del valor. Un nuevo valor redibuja solo la capa dinámica, de modo que un panel con muchos medidores sigue siendo rápido mientras cambian los valores.
sgcGauges forma parte de la suite visual de eSeGeCe. Se dibuja con sgcUICore, el paquete de runtime compartido con el canvas, el rasterizador, las fuentes, el escritor de SVG y los temas, de modo que los medidores combinan con los demás productos de la suite. sgcGauges requiere sgcUICore, que se instala junto con él. El puente sgcGaugesWS de la edición Registered alimenta los medidores con mensajes MQTT, WebSocket y eventos enviados por el servidor, a través de sgcWebSockets.

Ediciones

sgcGauges se distribuye como Registered, Trial y Community. Community conserva los medidores radial, lineal, digital (7 segmentos), LED, barra de LEDs y anillo de progreso, con rangos, umbrales, animación, temas y la salida PNG y SVG sin ventana, para VCL y FireMonkey. Deja fuera el VU meter, el reloj, la brújula y el anunciador, las escalas y los punteros posteriores al primero, las pantallas de 14 segmentos y de matriz de puntos y el puente. Véase Ediciones.

Por dónde empezar

- Arquitectura: los modelos, los hosts, las dos capas de renderizado y los hilos.
- Tipos de medidor: los diez medidores, sus clases y units de un vistazo.
- Instalación y Plataformas: el instalador, los paquetes, las versiones de Delphi y C++Builder y las plataformas.
- Inicio rápido VCL, Inicio rápido FMX, Inicio rápido sin ventana e Inicio rápido C++Builder: un primer medidor en pocos minutos.
- Componentes: una página por control, desde el medidor radial hasta el anunciador, y las clases de modelo TsgcCustomGauge, TsgcGaugeScale y TsgcGaugePointer.
- Escalas, Rangos, Punteros, Umbrales, Animación, Temas, Salida sin ventana y Ediciones.
- Puente: alimente medidores con mensajes MQTT, WebSocket y eventos enviados por el servidor.
- Solución de problemas: problemas habituales y sus causas.

Arquitectura

sgcGauges separa lo que es un medidor del lugar donde se muestra. Una clase de modelo guarda la configuración y los valores y se dibuja en un canvas de sgcUICore. Un control host coloca un modelo en un formulario VCL o FireMonkey.

Modelos

Cada tipo de medidor es una clase que desciende de **TsgcCustomGauge** (unit sgcGauges_Model), como TsgcRadialGauge o TsgcLinearGauge. Un modelo es un TPersistent sin ventana, de modo que también se ejecuta en un programa de consola, en un servicio o en Linux64. Contiene:

- **Scales**: una colección de TsgcGaugeScale, cada una con un rango de valores, marcas, etiquetas y una colección de **Ranges** de color.
- **Pointers**: una colección de TsgcGaugePointer, cada uno mostrando un valor en una escala.
- **Thresholds**: una colección de TsgcGaugeThreshold, valores que lanzan OnThreshold cuando un puntero los cruza.
- **Animation**: la duración y el suavizado del movimiento de un puntero.
- El aspecto compartido: Theme, Title, TitleFont, Units, ShowValue, ValueFormat, ValueFont, BackgroundColor y FaceColor.

Un modelo nuevo siempre tiene una escala, de 0 a 100, y un puntero. **Value** es un atajo a Pointers[0].Value, y SetValue(aPointer, aValue, aAnimate) escribe el valor de cualquier puntero.

Hosts

Los controles VCL descienden de **TsgcVCLCustomGauge** (unit sgcGauges_VCL) y los controles FireMonkey de **TsgcFMXCustomGauge** (unit sgcGauges_FMX). Cada control crea el modelo de su tipo, que devuelve la propiedad pública **Gauge**, y publica las propiedades del modelo que su tipo dibuja. Un host se repinta cuando cambia el modelo, ejecuta los fotogramas de animación, entrega OnThreshold en el hilo principal y se guarda con SaveToPNG y SaveToSVG con su tamaño de cliente.

Dos capas de renderizado

Un modelo dibuja en dos capas:

- La **capa estática** (RenderStatic): el fondo, la esfera, el bisel, las escalas, las marcas, las etiquetas, los rangos, las marcas de umbral y el título. Un cambio de propiedad dispara OnChange con gckStatic e incrementa StaticStamp.
- La **capa dinámica** (RenderDynamic): los punteros y el texto del valor. Un nuevo valor dispara OnChange solo con gckDynamic.

Los hosts conservan la capa estática como píxeles en caché y dibujan solo la capa dinámica mientras cambia un valor o se mueve un puntero. BeginUpdate y EndUpdate reúnen varios cambios en un único OnChange, estático cuando alguno de los cambios fue estático. Render dibuja las dos capas a la vez.

Tamaños, ángulos y colores

- Los tamaños de escalas y punteros (Radius, Offset, Thickness, Width, Length) son fracciones del tamaño del medidor, de modo que un medidor tiene el mismo aspecto con cualquier tamaño. El texto también se adapta al medidor.
- Los ángulos están en grados, 0 son las 3 en punto y los ángulos positivos giran en sentido horario. El reloj y la brújula cuentan desde las 12 y desde el norte.
- Los colores son valores TsgcUIColor con la forma $AARRGGBB. Un color igual a 0 usa el color del tema.
- Los valores se limitan al rango de su escala cuando se escriben. NaN se ignora y un valor infinito va al límite más cercano.

Hilos

Value, SetValue y el Value y el EndValue de un puntero pueden escribirse desde cualquier hilo. En un modelo, OnChange y OnThreshold se llaman entonces en ese hilo. Los controles VCL y FireMonkey llaman a su OnThreshold en el hilo principal, sea cual sea el hilo que escribió el valor. Todas las demás propiedades están pensadas para el hilo principal.

Units

Unit	Contenido
`sgcGauges_Types`	Las enumeraciones y las funciones auxiliares.
`sgcGauges_Model`	TsgcCustomGauge, las escalas, rangos, punteros, umbrales y la animación.
`sgcGauges_Render`	El painter con el que dibuja cada tipo de medidor.
`sgcGauges_Radial`, `_Linear`, `_Digital`, `_LED`, `_ProgressRing`, `_VUMeter`, `_Clock`, `_Compass`, `_Annunciator`	Las clases de modelo de los diez tipos de medidor. El LED y la barra de LEDs comparten sgcGauges_LED.
`sgcGauges_VCL`	Los controles VCL.
`sgcGauges_FMX`	Los controles FireMonkey.
`sgcGaugesWS_Source`, `_Source_MQTT`, `_Source_WebSocket`, `_Source_SSE`	El puente sgcGaugesWS, solo en la edición Registered.

Véase también

- Tipos de medidor
- Salida sin ventana
- Animación

Tipos de medidor

sgcGauges tiene diez tipos de medidor. Cada uno es una clase de modelo en su propia unit, un control VCL en sgcGauges_VCL y un control FireMonkey en sgcGauges_FMX. Los controles están en la página **SGC Gauges** de la paleta.

Medidor	Clase de modelo (unit)	Controles VCL y FireMonkey	Edición	Qué muestra
Radial	TsgcRadialGauge (sgcGauges_Radial)	TsgcVCLRadialGauge, TsgcFMXRadialGauge	Community	Una esfera con arco completo, medio, de cuarto o personalizado, un bisel y el texto del valor. RadialStyle, StartAngle, SweepAngle, BezelStyle, ValuePosition.
Lineal	TsgcLinearGauge (sgcGauges_Linear)	TsgcVCLLinearGauge, TsgcFMXLinearGauge	Community	Una barra horizontal o vertical, un termómetro o un gráfico bullet con una línea de objetivo. Orientation, LinearStyle, Target, BulbSize.
Digital	TsgcDigitalGauge (sgcGauges_Digital)	TsgcVCLDigitalGauge, TsgcFMXDigitalGauge	Community (7 segmentos)	Una pantalla de segmentos de un valor o de un texto: 7 segmentos, 14 segmentos o una matriz de puntos 5 x 7. DisplayType, Digits, Text, Skew, SegmentColor, GhostColor, ShowGhost, Alignment.
LED	TsgcLEDGauge (sgcGauges_LED)	TsgcVCLLEDGauge, TsgcFMXLEDGauge	Community	Una lámpara encendida o apagada, con resplandor y parpadeo opcional. Active, OnColor, OffColor, Shape, Blink, BlinkInterval.
Barra de LEDs	TsgcLEDBarGauge (sgcGauges_LED)	TsgcVCLLEDBarGauge, TsgcFMXLEDBarGauge	Community	Una fila o columna de segmentos de LED con retención de pico. Segments, SegmentGap, Orientation, PeakHold, PeakHoldTime.
Anillo de progreso	TsgcProgressRingGauge (sgcGauges_ProgressRing)	TsgcVCLProgressRingGauge, TsgcFMXProgressRingGauge	Community	Un anillo que se rellena con el valor, con un porcentaje en el centro. Thickness, TrackColor, RoundCaps, ShowPercent.
VU meter	TsgcVUMeterGauge (sgcGauges_VUMeter)	TsgcVCLVUMeterGauge, TsgcFMXVUMeterGauge	Registered	Un medidor de nivel de audio con una aguja o una barra y retención de pico.
Reloj	TsgcClockGauge (sgcGauges_Clock)	TsgcVCLClockGauge, TsgcFMXClockGauge	Registered	Un reloj analógico, en vivo o a una hora dada, con desfase UTC opcional.

Brújula	TsgcCompassGauge (sgcGauges_Compass)	TsgcVCLCompassGauge, TsgcFMXCompassGauge	Registered	Una brújula con rosa giratoria o aguja giratoria y un rumbo objetivo.
Anunciador	TsgcAnnunciatorGauge (sgcGauges_Annunciator)	TsgcVCLAnnunciatorGauge, TsgcFMXAnnunciatorGauge	Registered	Una cuadrícula de casillas de alarma en cuatro estados, que parpadean hasta que se reconocen.

Lo que comparte todo medidor

Todo tipo desciende de TsgcCustomGauge y por tanto tiene Scales, Pointers, Thresholds, Animation, Theme y las propiedades del texto del valor, pero cada control publica solo lo que dibuja su tipo. El LED, por ejemplo, no tiene escala, el reloj no tiene Value y la brújula publica Heading en lugar de Value. Units se dibuja junto al texto del valor, y el texto del valor usa ValueFormat o, cuando está vacío, los decimales de la escala.

Véase también

- Arquitectura
- Ediciones
- Inicio rápido VCL

Instalación

sgcGauges forma parte de la distribución sgcUI. Un único instalador cubre los productos sgcUI e instala solo los productos que incluye su licencia.

Instalación con el instalador de sgcUI

1. Ejecute el instalador de sgcUI e inicie sesión con el nombre de usuario y la contraseña de su cuenta de eSeGeCe.
2. Elija las versiones de Delphi y C++Builder en las que instalar.
3. En la página de componentes, seleccione **sgcGauges** y los demás productos que desee. El servidor de licencias devuelve los productos que usted posee, y el instalador muestra una casilla por producto. Los productos que su licencia no incluye aparecen atenuados. Una licencia All-Access desbloquea todos los productos.
4. El instalador copia los archivos, después compila los paquetes en su máquina para cada IDE seleccionado, en orden de dependencia, e instala los paquetes de tiempo de diseño.

Qué acompaña a sgcGauges

sgcUICore, la base compartida de los productos visuales sgcUI, acompaña a sgcGauges. No tiene casilla propia. El instalador lo instala automáticamente siempre que se selecciona sgcGauges.

Carpetas instaladas

Cada producto se instala en una carpeta, por defecto `Documents\esegece\sgcUI` (`sgcUI_Trial` para la versión de prueba y `sgcUI_Community` para la edición Community).

Carpeta	Contenido
`Source`	El código fuente de todos los productos instalados. Esta es la única ruta de biblioteca de sgcUI.
`Packages`	Los paquetes de todos los productos instalados, un conjunto por producto y versión del IDE.
`Lib<V>`	Las units compiladas para cada versión del IDE y plataforma, por ejemplo `LibD13` .
`Demos\Gauges`	Los demos de sgcGauges: Gallery, GalleryFMX, Headless, HMIPanel, DashboardGauges y MQTTLive, para Delphi y C++Builder.

El instalador añade `sgcUI\Source` a la ruta de biblioteca y a la ruta de exploración de cada IDE y plataforma seleccionados, junto con la carpeta de las units compiladas. sgcUICore está en la misma carpeta `Source`, de modo que no necesita una ruta propia.

La paleta de componentes

Los componentes están en la página **SGC Gauges** de la paleta. Un formulario VCL muestra los controles VCL y un formulario FireMonkey los controles FMX. Las ediciones Registered y Trial registran 20 controles, los diez tipos de medidor para cada framework, y el puente añade sus tres fuentes de alimentación (TsgcGaugeFeedSource_WebSocket, TsgcGaugeFeedSource_MQTT y TsgcGaugeFeedSource_SSE). Community registra 12 controles, los seis tipos de medidor de Community para cada framework.

Trial y Community

- **Trial**: el instalador de prueba de sgcUI instala todos los productos en modo de prueba, sgcGauges incluido.
- **Community**: el instalador gratuito sgcUI Community incluye sgcGauges sin sus características avanzadas. Véase Ediciones.

Compilación manual de los paquetes

Para compilar los paquetes usted mismo, ábralos para su versión del IDE desde `sgcUI\Packages`. `<V>` es el sufijo de la versión de Delphi, por ejemplo `sgcGaugesD13.dpk`.

Paquete	Contenido
`sgcUICore<V>` , `sgcUICoreVCL<V>` , `sgcUICoreFMX<V>`	La base compartida sgcUICore y sus partes VCL y FireMonkey (runtime).
`sgcGauges<V>`	Las units independientes del framework: los tipos, el modelo, el painter y los diez modelos de medidor (runtime).
`sgcGaugesVCL<V>` / `dclsgcGauges<V>`	Los controles VCL (runtime / tiempo de diseño).
`sgcGaugesFMX<V>` / `dclsgcGaugesFMX<V>`	Los controles FireMonkey, desde Delphi XE6 (runtime / tiempo de diseño).
`sgcGaugesWS<V>` / `dclsgcGaugesWS<V>`	El puente hacia sgcWebSockets, solo en la edición Registered. Requiere el paquete sgcWebSockets.

Compílelos en orden de dependencia: `sgcUICore`, `sgcUICoreVCL` y `sgcUICoreFMX`, después `sgcGauges`, `sgcGaugesVCL` y `sgcGaugesFMX`, después los paquetes de tiempo de diseño. El grupo de proyectos `sgcGauges<V>.groupproj` contiene los paquetes de sgcGauges en ese orden. A continuación, instale los paquetes de tiempo de diseño y añada `sgcUI\Source` a la ruta de biblioteca de su versión de Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para C++Builder, abra los proyectos `.cbproj` que correspondan a su versión, por ejemplo `sgcGaugesC13.cbproj` y `sgcGaugesVCLC13.cbproj`, y compílelos. Las cabeceras `.hpp` generadas aparecen junto al paquete compilado.

Véase también

- Plataformas
- Inicio rápido VCL

Plataformas

Destino	Compatibilidad
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 y 13, con los proyectos de paquete sgcGaugesC2007 a sgcGaugesC13. Los paquetes generan las cabeceras .hpp y los archivos .lib que un proyecto de C++Builder incluye y enlaza. Véase Inicio rápido C++Builder.
VCL	Los controles TsgcVCL en Windows, Win32 y Win64, en todas las versiones de Delphi.
FireMonkey	Los controles TsgcFMX desde Delphi XE6 (SGC_GAUGES_FMX), en las plataformas FireMonkey. Usan solo FireMonkey y sgcUICore.
Linux64	Las units independientes del framework desde Delphi 10.2 (SGC_GAUGES_LINUX): el modelo, el painter y los diez modelos de medidor, con la salida PNG y SVG sin ventana. No hay controles para Linux.
.NET	Una versión en C# con la misma API: los diez medidores como controles WinForms, WPF y Avalonia y la salida PNG y SVG sin ventana, en .NET Framework 4.7.2 y .NET 8 y 9, véase Descripción general de .NET.

Paquetes

sgcGauges incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcGaugesD13.dpk para Delphi 13 y sgcGaugesC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcGauges	Runtime, independiente del framework	Todas las unidades del producto que no usan ni la VCL ni FireMonkey: el modelo, el painter y los diez modelos de medidor, con la salida PNG y SVG sin ventana.	rtl, sgcUICore
sgcGaugesVCL	Runtime, VCL	sgcGauges_VCL (TsgcVCLRadialGauge, TsgcVCLLinearGauge, TsgcVCLDigitalGauge, TsgcVCLLEDGauge, TsgcVCLLEDBarGauge, TsgcVCLProgressRingGauge, TsgcVCLVUMeterGauge, TsgcVCLClockGauge, TsgcVCLCompassGauge y TsgcVCLAnnunciatorGauge)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcGauges
sgcGaugesFMX	Runtime, FireMonkey, desde XE6	sgcGauges_FMX (TsgcFMXRadialGauge, TsgcFMXLinearGauge, TsgcFMXDigitalGauge, TsgcFMXLEDGauge, TsgcFMXLEDBarGauge, TsgcFMXProgressRingGauge, TsgcFMXVUMeterGauge, TsgcFMXClockGauge, TsgcFMXCompassGauge y TsgcFMXAnnunciatorGauge)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcGauges
dclsgcGauges	Tiempo de diseño, VCL	Registra los medidores VCL en la página de paleta **SGC Gauges** (TsgcVCLVUMeterGauge, TsgcVCLClockGauge, TsgcVCLCompassGauge y TsgcVCLAnnunciatorGauge no en la edición Community), con el editor de colecciones para Scales, Pointers, Thresholds y Tiles.	designide, rtl, vcl, sgcGauges, sgcGaugesVCL

dclsgcGaugesFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra los medidores FireMonkey en la página de paleta **SGC Gauges**.	designide, rtl, fmx, sgcGauges, sgcGaugesFMX, dclsgcGauges
sgcGaugesWS, dclsgcGaugesWS	Runtime y tiempo de diseño, puente de sgcWebSockets	Las fuentes de datos en vivo (TsgcGaugeFeedSource_WebSocket, TsgcGaugeFeedSource_MQTT y TsgcGaugeFeedSource_SSE), independientes del framework, en la página de paleta **SGC Gauges**.	rtl, sgcGauges, sgcWebSockets

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcGaugesFMX requiere dclsgcGauges, así que instale primero dclsgcGauges. Ninguna unit de sgcGauges usa una unit de sgcWebSockets, salvo las units del puente opcional sgcGaugesWS.

Versiones del IDE y plataformas de destino

Delphi	sgcGauges	sgcGaugesVCL	sgcGaugesFMX	dclsgcGauges, dclsgcGaugesFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE. sgcGaugesWS se compila para las mismas plataformas que sgcGauges, y dclsgcGaugesWS para las mismas plataformas que dclsgcGauges.

C++Builder	sgcGauges, sgcGaugesVCL	sgcGaugesFMX	dclsgcGauges, dclsgcGaugesFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcGauges)

XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcGauges)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcGaugesC13.cbproj, sgcGaugesVCLC13.cbproj, sgcGaugesFMXC13.cbproj, dclsgcGaugesC13.cbproj y dclsgcGaugesFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. sgcGaugesWS y dclsgcGaugesWS se compilan para las mismas versiones y plataformas de C++Builder que sgcGauges y dclsgcGauges, excepto Win64x, porque necesitan una compilación Win64x del paquete sgcWebSockets. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXRadialGauge, TsgcFMXLinearGauge, TsgcFMXDigitalGauge, TsgcFMXLEDGauge, TsgcFMXLEDBarGauge, TsgcFMXProgressRingGauge, TsgcFMXVUMeterGauge, TsgcFMXClockGauge, TsgcFMXCompassGauge y TsgcFMXAnnunciatorGauge) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcGaugesVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcGauges_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcGauges se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa el modelo, el painter y los diez modelos de medidor, con la salida PNG y SVG sin ventana. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Restricciones de versión

- Las units independientes del framework y las units VCL compilan en Delphi 7: sin genéricos, sin métodos anónimos, sin variables inline y sin record helpers.
- La salida es la misma en Win32, Win64 y Linux64: los archivos PNG y SVG de un medidor son idénticos byte a byte en las tres plataformas.
- Las enumeraciones tienen prefijos de medidor (grs, gpk, gdt, gtd, ge y así sucesivamente), de modo que las cabeceras de C++Builder nunca chocan con los demás productos sgcUI.

Véase también

- Ediciones
- Arquitectura

Inicio rápido VCL

Instale los paquetes de sgcGauges para su versión de Delphi. Los componentes están en la página **SGC Gauges** de la paleta.

En tiempo de diseño

1. Coloque un **TsgcVCLRadialGauge** en un formulario. Escriba Speed en Title y km/h en Units.
2. Abra la colección Scales, seleccione la primera escala y defina Maximum en 260 y MajorInterval en 20. Abra su colección Ranges, añada un rango y defina StartValue en 200, EndValue en 260 y Color en un rojo como $FFE34948.
3. Coloque un **TsgcVCLLEDGauge** a su lado, defina OnColor con el mismo rojo y Blink en True.
4. Añada un umbral a la colección Thresholds del medidor radial con Value 200, Hysteresis 10 y Direction gtd-Both, y escriba un manejador OnThreshold que asigne aRising a la propiedad Active del LED.
5. Defina Value en código, desde un temporizador o desde un hilo de trabajo, y ejecute la aplicación. La aguja se mueve a cada nuevo valor en 300 ms.

En el código

Un medidor funciona en cuanto tiene un Parent. Defina el rango de una escala antes del valor, porque un valor se limita a la escala cuando se escribe.

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

type
  TSpeedPanel = class(TComponent)
  public
    Speed: TsgcVCLRadialGauge;
    Alarm: TsgcVCLLEDGauge;
    procedure SpeedThreshold(Sender: TObject; aThreshold: TsgcGaugeThreshold;
      aValue: Double; aRising: Boolean);
  end;

procedure TSpeedPanel.SpeedThreshold(Sender: TObject;
  aThreshold: TsgcGaugeThreshold; aValue: Double; aRising: Boolean);
begin
  // called in the main thread, whatever thread wrote the value
  Alarm.Active := aRising;
end;

function CreateSpeedPanel(aForm: TForm): TSpeedPanel;
var
  oRange: TsgcGaugeRange;
  oThreshold: TsgcGaugeThreshold;
begin
  Result := TSpeedPanel.Create(aForm);

  Result.Speed := TsgcVCLRadialGauge.Create(aForm);
  Result.Speed.Parent := aForm;
  Result.Speed.SetBounds(16, 16, 240, 240);
  Result.Speed.Title := 'Speed';
  Result.Speed.Units := 'km/h';
  // the default scale is 0..100
  Result.Speed.Scales[0].Maximum := 260;
  Result.Speed.Scales[0].MajorInterval := 20;
  // a red band at the end of the scale
  oRange := Result.Speed.Scales[0].Ranges.Add;
  oRange.StartValue := 200;
```

```
  oRange.EndValue := 260;
  oRange.Color := $FFE34948;
  // on at 200 km/h, off again below 190 km/h
  oThreshold := Result.Speed.Thresholds.Add;
  oThreshold.Value := 200;
  oThreshold.Hysteresis := 10;
  oThreshold.Direction := gtdBoth;
  Result.Speed.OnThreshold := Result.SpeedThreshold;

  Result.Alarm := TsgcVCLLEDGauge.Create(aForm);
  Result.Alarm.Parent := aForm;
  Result.Alarm.SetBounds(272, 16, 48, 48);
  Result.Alarm.OnColor := $FFE34948;
  Result.Alarm.Blink := True;

  // the needle moves from 0 to 120 in 300 ms
  Result.Speed.Value := 120;
end;
```

Véase también

- Tipos de medidor
- Umbrales
- Animación

Inicio rápido FMX

Los controles FireMonkey tienen las mismas propiedades que los VCL, con el prefijo TsgcFMX, desde Delphi XE6. Instale los paquetes de sgcGauges, abra un formulario FireMonkey y use los componentes de la página **SGC Gauges** de la paleta.

En tiempo de diseño

1. Coloque un **TsgcFMXRadialGauge** en un formulario, defina RadialStyle como grsHalf y Theme como utDark.
2. Defina Title, Units y ValueFormat, y el Maximum de la primera escala en la colección Scales.
3. Coloque un **TsgcFMXLinearGauge**, defina Orientation como gloVertical y LinearStyle como glsThermometer.
4. Ejecute la aplicación en cualquier plataforma FireMonkey.

Qué tener en cuenta en FireMonkey

- Los controles dibujan a la escala de la escena, de modo que se ven nítidos en pantallas de DPI alto y en dispositivos móviles.
- OnThreshold se llama en el hilo principal mediante TThread.Queue, sea cual sea el hilo que escribió el valor.

En el código

```
uses
  System.SysUtils, System.Classes, FMX.Types, FMX.Forms,
  sgcUICore_Theme, sgcGauges_Types, sgcGauges_Model, sgcGauges_FMX;

type
  TBoilerPanel = class(TComponent)
  public
    Pressure: TsgcFMXRadialGauge;
    Level: TsgcFMXLinearGauge;
    procedure Update(aPressure, aLevel: Double);
  end;

procedure TBoilerPanel.Update(aPressure, aLevel: Double);
begin
  // Value may be written from any thread
  Pressure.Value := aPressure;
  Level.Value := aLevel;
end;

function CreateBoilerPanel(aForm: TForm): TBoilerPanel;
begin
  Result := TBoilerPanel.Create(aForm);

  Result.Pressure := TsgcFMXRadialGauge.Create(aForm);
  Result.Pressure.Parent := aForm;
  Result.Pressure.SetBounds(16, 16, 240, 160);
  Result.Pressure.RadialStyle := grsHalf;
  Result.Pressure.Theme := utDark;
  Result.Pressure.Title := 'Pressure';
  Result.Pressure.Units := 'bar';
  Result.Pressure.ValueFormat := '0.0';
  Result.Pressure.Scales[0].Maximum := 10;

  Result.Level := TsgcFMXLinearGauge.Create(aForm);
  Result.Level.Parent := aForm;
```

```
  Result.Level.SetBounds(272, 16, 80, 240);
  Result.Level.Orientation := gloVertical;
  Result.Level.LinearStyle := glsThermometer;
  Result.Level.Theme := utDark;
  Result.Level.Units := '%';

  Result.Update(6.5, 72);
end;
```

Véase también

- Inicio rápido VCL
- Temas
- Plataformas

Inicio rápido sin ventana

Las clases de modelo se renderizan sin ventana. Un programa de consola, un servidor web o un informe puede crear un medidor, definir sus valores y guardarlo como archivo PNG o SVG. Las units del modelo no usan VCL ni FireMonkey, de modo que el mismo código se ejecuta en Windows y en Linux64.

Pasos

1. Añada sgcGauges_Model y la unit del tipo de medidor a la cláusula uses, por ejemplo sgcGauges_Radial.
2. Cree el modelo, por ejemplo `TsgcRadialGauge.Create`, y defina sus escalas, rangos y valor como en un control.
3. Llame a SaveToPNG o SaveToSVG con el tamaño en píxeles lógicos y una escala opcional, y después libere el modelo.

En el código

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Radial;

procedure SaveSpeedGauge(const aFolder: string; aSpeed: Double);
var
  oGauge: TsgcRadialGauge;
  oRange: TsgcGaugeRange;
begin
  oGauge := TsgcRadialGauge.Create;
  try
    oGauge.Title := 'Speed';
    oGauge.Units := 'km/h';
    oGauge.Scales[0].Maximum := 260;
    oGauge.Scales[0].MajorInterval := 20;
    oRange := oGauge.Scales[0].Ranges.Add;
    oRange.StartValue := 200;
    oRange.EndValue := 260;
    // a headless render always draws the target value, never a frame of a move
    oGauge.Value := aSpeed;
    // 240 x 240 logical pixels; the PNG at scale 2 is 480 x 480 pixels
    oGauge.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'speed.png',
      240, 240, 2);
    oGauge.SaveToSVG(IncludeTrailingPathDelimiter(aFolder) + 'speed.svg',
      240, 240);
  finally
    oGauge.Free;
  end;
end;
```

Un renderizado sin ventana dibuja los valores de destino. La animación y la temporización de los fotogramas pertenecen a los controles. Véase Salida sin ventana.

Véase también

- Salida sin ventana
- Arquitectura

Inicio rápido C++Builder

Instale los paquetes de sgcGauges para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las units compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de include y de biblioteca del proyecto, y use los componentes de la página **SGC Gauges** de la paleta exactamente igual que en Delphi. El instalador añade las dos rutas al IDE por usted. Las cabeceras .hpp y los archivos .lib generados de sgcGauges y sgcUICore están en esa única carpeta.

Qué tener en cuenta en C++

- Incluya la cabecera de la unit que declara un control: `sgcGauges_VCL.hpp` para los controles VCL y `sgcGauges_FMX.hpp` para los de FireMonkey. Las colecciones y los tipos de evento están en `sgcGauges_Model.hpp` y las enumeraciones en `sgcGauges_Types.hpp`.
- Las colecciones no tienen índice por defecto en C++: escriba `Gauge->Scales->Items[0]` donde Delphi escribe `Gauge.Scales[0]`.
- Los eventos son closures, de modo que sus manejadores son funciones miembro __fastcall de una clase.
- Los colores son valores de 32 bits sin signo con la forma 0xAARRGGBB.
- Las clases de modelo tienen constructores sin argumentos, como `new TsgcRadialGauge()`, y usted mismo las elimina con delete. Véase Salida sin ventana.

En el código

Véase también

- Inicio rápido VCL
- Instalación

Medidor radial

Clases: TsgcVCLRadialGauge (unit sgcGauges_VCL), TsgcFMXRadialGauge (unit sgcGauges_FMX) y el modelo TsgcRadialGauge (unit sgcGauges_Radial)
Una esfera: las escalas sobre un arco, un bisel alrededor de la cara, los punteros y el texto del valor. El arco y su tamaño se ajustan al control, de modo que un medio medidor tiene la parte inferior plana y llena un control ancho.
El medidor forma parte de todas las ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
RadialStyle	TsgcGaugeRadialStyle	El arco: grsFull (por defecto, desde 135 grados a lo largo de 270), grsHalf (180 a lo largo de 180), grsQuarter (180 a lo largo de 90) o grsCustom.
StartAngle	Double	El inicio del arco con grsCustom, en grados, 0 son las 3 en punto y positivo es en sentido horario. Por defecto 135.
SweepAngle	Double	La longitud del arco con grsCustom, en grados. Por defecto 270.
BezelStyle	TsgcGaugeBezelStyle	gbzNone, gbzThin (por defecto) o gbzThick.
ValuePosition	TsgcGaugeValuePosition	Dónde va el texto del valor: gvpBottom (por defecto), gvpCenter o gvpTop.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Los valores que lanzan OnThreshold cuando un puntero los cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
Units	string	El texto dibujado junto al texto del valor, como km/h.
ShowValue	Boolean	Dibuja el texto del valor. Por defecto True.
ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales de la escala.

ValueFont	TsgcUITextStyle	La fuente del texto del valor.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) reemplaza el texto del valor.

Ejemplo

Un medio tacómetro con un rango rojo:

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateTachometer(aForm: TForm): TsgcVCLRadialGauge;
var
  oRange: TsgcGaugeRange;
begin
  Result := TsgcVCLRadialGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 260, 200);
  Result.RadialStyle := grsHalf;
  Result.BezelStyle := gbzThick;
  Result.ValuePosition := gvpCenter;
  Result.Title := 'RPM';
  Result.Units := 'x1000';
  Result.ValueFormat := '0.0';
  Result.Scales[0].Maximum := 8;
  Result.Scales[0].MajorInterval := 1;
  oRange := Result.Scales[0].Ranges.Add;
  oRange.StartValue := 6.5;
  oRange.EndValue := 8;
  oRange.Color := $FFE34948;
  Result.Value := 3.2;
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

Medidor lineal

Clases: TsgcVCLLinearGauge (unit sgcGauges_VCL), TsgcFMXLinearGauge (unit sgcGauges_FMX) y el modelo TsgcLinearGauge (unit sgcGauges_Linear)
Un medidor horizontal o vertical dibujado como barra, termómetro o gráfico bullet. El puntero por defecto es una barra.
El medidor forma parte de todas las ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Orientation	TsgcGaugeOrientation	gloHorizontal (por defecto) va de izquierda a derecha, gloVertical de abajo arriba.
LinearStyle	TsgcGaugeLinearStyle	glsBar (por defecto): la pista de la escala y los punteros. glsThermometer: la escala 0 es un tubo con un bulbo en el mínimo, y el puntero 0 se dibuja como el líquido. glsBullet: bandas gruesas, los rangos detrás de una barra más fina y una línea Target.
Target	Double	La línea de objetivo del estilo bullet, dibujada a través de la escala 0 cuando queda dentro de la escala. Por defecto 0.
BulbSize	Double	El radio del bulbo del termómetro como fracción del tamaño del medidor. Por defecto 0.06.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Los valores que lanzan OnThreshold cuando un puntero los cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
Units	string	El texto dibujado junto al texto del valor, como km/h.
ShowValue	Boolean	Dibuja el texto del valor. Por defecto True.

ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales de la escala.
ValueFont	TsgcUITextStyle	La fuente del texto del valor.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) reemplaza el texto del valor.

Ejemplo

Un termómetro vertical y un gráfico bullet:

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateThermometer(aForm: TForm): TsgcVCLLinearGauge;
begin
  Result := TsgcVCLLinearGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 90, 280);
  Result.Orientation := gloVertical;
  Result.LinearStyle := glsThermometer;
  Result.BulbSize := 0.08;
  Result.Title := 'Oven';
  Result.Units := 'C';
  Result.Scales[0].Minimum := 0;
  Result.Scales[0].Maximum := 250;
  Result.Value := 180;
end;

function CreateBullet(aForm: TForm): TsgcVCLLinearGauge;
begin
  Result := TsgcVCLLinearGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(120, 16, 320, 90);
  Result.LinearStyle := glsBullet;
  Result.Title := 'Sales';
  Result.Target := 80; // the target line
  Result.Value := 64;
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

Medidor digital

Clases: TsgcVCLDigitalGauge (unit sgcGauges_VCL), TsgcFMXDigitalGauge (unit sgcGauges_FMX) y el modelo TsgcDigitalGauge (unit sgcGauges_Digital)
Una pantalla de segmentos para el valor del puntero 0, o para un texto. El tipo de 7 segmentos muestra dígitos, el signo menos, el punto decimal y los dos puntos; el tipo de 14 segmentos añade letras; la matriz de puntos 5 x 7 muestra ASCII 32 a 126. Los segmentos apagados pueden dibujarse tenues detrás de los encendidos.
El número de posiciones de glifo sale de Text, o del más ancho de los textos de Minimum y Maximum de la escala, salvo que Digits lo fije. Un valor que no cabe muestra guiones, y un Text que no cabe se recorta tras Digits glifos. En los tipos de segmentos, un '.' o ',' enciende el punto decimal del glifo anterior y no ocupa posición propia.
La pantalla de 7 segmentos forma parte de todas las ediciones; en Community los tipos de 14 segmentos y de matriz de puntos se dibujan como 7 segmentos. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
DisplayType	TsgcGaugeDisplayType	gdt7Segment (por defecto), gdt14Segment o gdtDotMatrix.
Digits	Integer	El número de posiciones de glifo. 0 (por defecto) toma el número del texto.
Text	string	Un texto mostrado en lugar del valor. Vacío (por defecto) muestra el valor.
Skew	Double	Inclina los glifos, en grados de -30 a 30. Positivo mueve la parte superior hacia la derecha. Por defecto 0.
SegmentColor	TsgcUIColor	El color de los segmentos encendidos. 0 usa el color de ValueFont, si no el color de texto del tema. El ValueColor de un umbral cruzado tiene prioridad sobre él.
GhostColor	TsgcUIColor	El color de los segmentos apagados. 0 usa el color de los encendidos, tenue.
ShowGhost	Boolean	Dibuja los segmentos apagados. Por defecto True.
Alignment	TsgcUITextHAlign	La posición de un texto más corto. Por defecto thaRight.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Los valores que lanzan OnThreshold cuando un puntero los cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.

Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
Units	string	El texto dibujado junto al texto del valor, como km/h.
ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales de la escala.
ValueFont	TsgcUITextStyle	La fuente del texto del valor.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) reemplaza el texto del valor.

Ejemplo

Un contador de seis dígitos en verde y un texto fijo:

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateCounter(aForm: TForm): TsgcVCLDigitalGauge;
begin
  Result := TsgcVCLDigitalGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 280, 100);
  Result.DisplayType := gdt7Segment;
  Result.Digits := 6;
  Result.ValueFormat := '0.00';
  Result.Skew := 8;
  Result.SegmentColor := $FF2ECC71;
  Result.Scales[0].Maximum := 1000;
  Result.Value := 245.5;
end;

procedure ShowText(aGauge: TsgcVCLDigitalGauge);
begin
  // a Text that is not empty is shown instead of the value
  aGauge.Text := '12:45';
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

LED

Clases: TsgcVCLLEDGauge (unit sgcGauges_VCL), TsgcFMXLEDGauge (unit sgcGauges_FMX) y el modelo TsgcLEDGauge (unit sgcGauges_LED)
Una lámpara indicadora con resplandor, que puede parpadear. La lámpara está encendida mientras Active es True. Active es una vista de Value <> 0, de modo que un enlace de alimentación que escribe 0 o 1 en el valor enciende y apaga la lámpara. El LED conserva cualquier valor salvo NaN, sin limitarlo a una escala, y nunca se anima.
El medidor forma parte de todas las ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Active	Boolean	Enciende la lámpara. Al asignarla se escribe 1 o 0 en el valor. No se guarda en el formulario.
OnColor	TsgcUIColor	El color de la lámpara encendida. 0 es el verde de la paleta del tema.
OffColor	TsgcUIColor	El color de la lámpara apagada. 0 usa el color de la encendida, translúcido.
Shape	TsgcGaugeLEDShape	gshCircle (por defecto), gshSquare o gshRect.
Blink	Boolean	La lámpara encendida se enciende y se apaga cada BlinkInterval. Por defecto False. No hay parpadeo en tiempo de diseño.
BlinkInterval	Integer	El periodo de parpadeo en milisegundos. Por defecto 500, como mínimo 50.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Ejemplo

Una lámpara de alarma roja que parpadea mientras está encendida:

```
uses
  Classes, Controls, Forms,
```

```
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateAlarmLamp(aForm: TForm): TsgcVCLLEDGauge;
begin
  Result := TsgcVCLLEDGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 80, 100);
  Result.Title := 'Alarm';
  Result.Shape := gshCircle;
  Result.OnColor := $FFE34948;
  Result.Blink := True;
  Result.BlinkInterval := 400;
end;

procedure SetAlarm(aLamp: TsgcVCLLEDGauge; aOn: Boolean);
begin
  // Active is Value <> 0, so a feed may also write 0 or 1 to Value
  aLamp.Active := aOn;
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge
- Barra de LEDs

Barra de LEDs

Clases: TsgcVCLLEDBarGauge (unit sgcGauges_VCL), TsgcFMXLEDBarGauge (unit sgcGauges_FMX) y el modelo TsgcLEDBarGauge (unit sgcGauges_LED)
Una fila o una columna de segmentos de LED que dividen la escala del puntero 0. Un segmento está encendido cuando su centro queda entre el origen del puntero y su valor. Un segmento toma el color del rango de la escala en que se encuentra, si no verde, ámbar y rojo por tercios de la escala. La retención de pico mantiene encendido el segmento más alto durante un tiempo después de que el nivel baje.
El medidor forma parte de todas las ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Segments	Integer	El número de segmentos, de 1 a 100. Por defecto 10.
SegmentGap	Double	El espacio entre dos segmentos como fracción de un segmento, hasta 0.9. Por defecto 0.2.
Orientation	TsgcGaugeOrientation	gloHorizontal (por defecto) de izquierda a derecha, gloVertical de abajo arriba.
PeakHold	Boolean	Mantiene encendido el segmento de pico. Por defecto False.
PeakHoldTime	Integer	Cuánto tiempo se retiene el pico, en milisegundos. Por defecto 1000.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Ejemplo

Una barra de nivel vertical de 20 segmentos con un rango azul y retención de pico:

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateLevelBar(aForm: TForm): TsgcVCLLEDBarGauge;
var
  oRange: TsgcGaugeRange;
begin
  Result := TsgcVCLLEDBarGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 60, 260);
  Result.Orientation := gloVertical;
  Result.Segments := 20;
  Result.SegmentGap := 0.3;
  Result.PeakHold := True;
  Result.PeakHoldTime := 1500;
  // a range colours the segments inside it, the others keep the
  // green, amber and red thirds
  oRange := Result.Scales[0].Ranges.Add;
  oRange.StartValue := 0;
  oRange.EndValue := 60;
  oRange.Color := $FF2A78D6;
  Result.Value := 72;
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge
- Rangos

Anillo de progreso

Clases: TsgcVCLProgressRingGauge (unit sgcGauges_VCL), TsgcFMXProgressRingGauge (unit sgcGauges_FMX) y el modelo TsgcProgressRingGauge (unit sgcGauges_ProgressRing)
Un anillo que se rellena en sentido horario desde las 12 con el valor del puntero 0, con el porcentaje en el centro. La barra toma el Color del puntero, si no el color del rango que contiene el valor, si no el color de acento del tema. El puntero por defecto es una barra.
El medidor forma parte de todas las ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Thickness	Double	El ancho del anillo como fracción del tamaño del medidor. Por defecto 0.15.
TrackColor	TsgcUIColor	El color del anillo vacío. 0 usa el tema.
RoundCaps	Boolean	Redondea los dos extremos de una barra más corta que el anillo completo. Por defecto True.
ShowPercent	Boolean	True (por defecto) muestra el porcentaje del valor dentro de la escala, con ValueFormat o '0' y un signo %. False muestra el texto del valor y Units.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Los valores que lanzan OnThreshold cuando un puntero los cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
Units	string	El texto dibujado junto al texto del valor, como km/h.
ShowValue	Boolean	Dibuja el texto del valor. Por defecto True.
ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales de la escala.
ValueFont	TsgcUITextStyle	La fuente del texto del valor.

BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) reemplaza el texto del valor cuando ShowPercent es False.

Ejemplo

El progreso de una subida, con el número de bytes como valor:

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateUploadRing(aForm: TForm): TsgcVCLProgressRingGauge;
begin
  Result := TsgcVCLProgressRingGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 160, 160);
  Result.Title := 'Upload';
  Result.Thickness := 0.12;
  Result.RoundCaps := True;
  Result.ShowPercent := True;
  // the bytes sent: the ring shows the percentage of the scale
  Result.Scales[0].Maximum := 5242880;
  Result.Value := 1310720; // 25%
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

VU meter

Clases: TsgcVCLVUMeterGauge (unit sgcGauges_VCL), TsgcFMXVUMeterGauge (unit sgcGauges_FMX) y el modelo TsgcVUMeterGauge (unit sgcGauges_VUMeter)
Un medidor de nivel de audio. La escala por defecto va de -20 a +3 dB con el mapeo en decibelios, de modo que una posición sigue la amplitud, con las marcas por fuera y un rango rojo desde 0 dB. El puntero se mueve en amplitud, como un medidor real. Mientras MajorInterval es 0, la escala muestra las marcas clásicas del VU.
El medidor necesita la edición Registered o Trial: en Community el modelo no dibuja nada y los controles no se instalan. Véase Ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
MeterStyle	TsgcGaugeVUMeterStyle	gvmNeedle (por defecto): una aguja en un arco de 90 grados. gvmBar: una barra horizontal sobre una franja con la pista y los rangos.
PeakHold	Boolean	Muestra el pico del nivel. Por defecto False.
PeakHoldTime	Integer	Cuánto tiempo se retiene el pico antes de caer, en milisegundos. Por defecto 1500.
PeakDecay	Double	La velocidad a la que cae el pico tras la retención, en dB por segundo. Por defecto 20. 0 descarta el pico de inmediato tras la retención.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Los valores que lanzan OnThreshold cuando un puntero los cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
Units	string	El texto dibujado junto al texto del valor, como km/h.
ShowValue	Boolean	Dibuja el texto del valor. Por defecto True.
ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales de la escala.
ValueFont	TsgcUITextStyle	La fuente del texto del valor.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.

FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.
Value	Double	El valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.

Métodos

ResetPeak borra el pico.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) reemplaza el texto del valor.

Ejemplo

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateVUMeter(aForm: TForm): TsgcVCLVUMeterGauge;
begin
  Result := TsgcVCLVUMeterGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 280, 180);
  Result.Title := 'Left';
  Result.MeterStyle := gvmNeedle;
  Result.PeakHold := True;
  Result.PeakHoldTime := 1500;
  Result.PeakDecay := 20;
  // the default scale runs from -20 to +3 dB with a red range above 0
  Result.Value := -6;
end;

procedure NewLevel(aMeter: TsgcVCLVUMeterGauge; aDB: Double);
begin
  aMeter.Value := aDB; // any thread
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

Reloj

Clases: TsgcVCLClockGauge (unit sgcGauges_VCL), TsgcFMXClockGauge (unit sgcGauges_FMX) y el modelo TsgcClockGauge (unit sgcGauges_Clock)
Un reloj analógico con 60 marcas de minuto, 12 numerales y las agujas de hora, minuto y segundo. Muestra Time o, con Live, la hora real: la hora local, o la hora UTC más UTCOffset minutos con UseUTCOffset. Un reloj en vivo se repinta cada segundo, o cada minuto sin ShowSeconds. Los numerales usan el LabelFont de la escala 0. El reloj no tiene Value ni punteros de valor.
El medidor necesita la edición Registered o Trial: en Community el modelo no dibuja nada y los controles no se instalan. Véase Ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Time	TDateTime	La hora mostrada mientras Live es False.
Live	Boolean	Muestra la hora real. Por defecto False.
UseUTCOffset	Boolean	Un reloj en vivo muestra la hora UTC más UTCOffset en lugar de la hora local. Por defecto False.
UTCOffset	Integer	El desfase respecto a UTC en minutos, como 60 o -300. Por defecto 0.
Numerals	TsgcGaugeClockNumerals	gcnArabic (por defecto), gcnRoman o gcnNone.
ShowSeconds	Boolean	Dibuja la aguja de los segundos. Por defecto True.
SmoothSeconds	Boolean	La aguja de los segundos se desliza en lugar de saltar. Por defecto False.
HourHandColor	TsgcUIColor	El color de la aguja de las horas. 0 usa el tema.
MinuteHandColor	TsgcUIColor	El color de la aguja de los minutos. 0 usa el tema.
SecondHandColor	TsgcUIColor	El color de la aguja de los segundos. 0 usa el tema.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
ValueFont	TsgcUITextStyle	La fuente del texto del valor.

Background-Color	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.

El modelo también tiene una propiedad pública TimeSource, una función que devuelve la hora real. Cuando se asigna, un reloj en vivo la lee en lugar del reloj del sistema, lo que da fotogramas fijos en las pruebas.

Ejemplo

Un reloj en vivo para UTC+5:30 con numerales romanos, y una hora fija:

```
uses
  Classes, Controls, Forms, SysUtils,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateWorldClock(aForm: TForm): TsgcVCLClockGauge;
begin
  Result := TsgcVCLClockGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 200, 200);
  Result.Title := 'UTC+5:30';
  Result.Live := True;
  Result.UseUTCOffset := True;
  Result.UTCOffset := 330; // minutes
  Result.Numerals := gcnRoman;
  Result.ShowSeconds := True;
  Result.SmoothSeconds := True;
  Result.SecondHandColor := $FFE34948;
end;

procedure ShowFixedTime(aClock: TsgcVCLClockGauge);
begin
  aClock.Live := False;
  aClock.Time := EncodeTime(10, 10, 30, 0);
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

Brújula

Clases: TsgcVCLCompassGauge (unit sgcGauges_VCL), TsgcFMXCompassGauge (unit sgcGauges_FMX) y el modelo TsgcCompassGauge (unit sgcGauges_Compass)
Un indicador de rumbo. Heading es el valor del puntero 0 en una escala de 0 a 360 grados, en sentido horario desde el norte. Da la vuelta en lugar de limitarse (450 es 90, -30 es 330), y un movimiento se anima por el arco más corto, de modo que de 359 a 1 gira 2 grados. La rosa tiene una marca cada 10 grados, los puntos cardinales y etiquetas de grados cada 30 grados, en el LabelFont de la escala 0. El texto del rumbo muestra grados enteros y un signo de grado, salvo que se defina ValueFormat.
El medidor necesita la edición Registered o Trial: en Community el modelo no dibuja nada y los controles no se instalan. Véase Ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Heading	Double	El rumbo en grados, el valor del puntero 0. Puede escribirse desde cualquier hilo. No se guarda en el formulario.
CompassStyle	TsgcGaugeCompassStyle	gcsRotatingCard (por defecto): la rosa gira y el rumbo queda bajo una marca fija en la parte superior. gcsRotatingNeedle: la rosa es fija y una aguja apunta al rumbo.
ShowCardinals	Boolean	Dibuja N, E, S, W y los puntos intercardinales. Por defecto True.
ShowDegrees	Boolean	Dibuja las etiquetas de grados de la rosa. Por defecto True. El texto del rumbo sigue a ShowValue.
TargetHeading	Double	El rumbo de la marca de objetivo. Por defecto 0.
ShowTarget	Boolean	Dibuja la marca de objetivo en el borde, en el color de acento del tema. Por defecto False.
Scales	TsgcGaugeScales	Las escalas con sus marcas, etiquetas y rangos de color. Véase TsgcGaugeScale.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer.
Animation	TsgcGaugeAnimation	Enabled, Duration y Easing del movimiento de un puntero.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
ShowValue	Boolean	Dibuja el texto del valor. Por defecto True.
ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales de la escala.

ValueFont	TsgcUITextStyle	La fuente del texto del valor.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) reemplaza el texto del rumbo.

Ejemplo

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateCompass(aForm: TForm): TsgcVCLCompassGauge;
begin
  Result := TsgcVCLCompassGauge.Create(aForm);
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 220, 220);
  Result.Title := 'Heading';
  Result.CompassStyle := gcsRotatingCard;
  Result.ShowCardinals := True;
  Result.ShowTarget := True;
  Result.TargetHeading := 270;
  Result.Heading := 350;
end;

procedure Turn(aCompass: TsgcVCLCompassGauge);
begin
  // from 350 the card turns 20 degrees through north, the shortest arc
  aCompass.Heading := 10;
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

Anunciador

Clases: TsgcVCLAnnunciatorGauge (unit sgcGauges_VCL), TsgcFMXAnnunciatorGauge (unit sgcGauges_FMX) y el modelo TsgcAnnunciatorGauge (unit sgcGauges_Annunciator)
Un panel de alarmas: una cuadrícula de casillas, rellenada fila por fila bajo el título. Cada casilla tiene un Caption, un State (gasNormal, gasAdvisory, gasWarning o gasAlarm) y un indicador Acknowledged, y toma el color de su estado. Una advertencia o una alarma no reconocida parpadea; un aviso informativo es fijo. Un cambio de State borra Acknowledged, de modo que cada nuevo estado necesita su propio reconocimiento. Los títulos usan el tamaño y el peso de ValueFont y se recortan al ancho de la casilla. El anunciador no tiene escalas, punteros ni umbrales.
El medidor necesita la edición Registered o Trial: en Community el modelo no dibuja nada y los controles no se instalan. Véase Ediciones. La clase VCL desciende de TsgcVCLCustomGauge y la clase FireMonkey de TsgcFMXCustomGauge. Ambas publican las propiedades de la tabla siguiente, Theme, OnThreshold y MaxFPS (el límite de fotogramas por segundo de la animación, por defecto 60), además de las habituales de su framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick y los eventos de ratón. La propiedad pública Gauge devuelve el modelo, y SaveToPNG y SaveToSVG guardan el control con su tamaño de cliente multiplicado por aScale.

Propiedades

Propiedad	Tipo	Descripción
Tiles	TsgcGaugeAnnunciatorTiles	Las casillas, elementos de TsgcGaugeAnnunciatorTile con Caption, State y Acknowledged.
Columns	Integer	Las casillas por fila. Por defecto 4.
FlashInterval	Integer	El periodo de parpadeo en milisegundos. Por defecto 500, como mínimo 50.
NormalColor	TsgcUIColor	El color de una casilla normal. 0 usa el color de la esfera del tema.
AdvisoryColor	TsgcUIColor	El color de una casilla informativa. 0 es azul.
WarningColor	TsgcUIColor	El color de una casilla de advertencia. 0 es ámbar.
AlarmColor	TsgcUIColor	El color de una casilla de alarma. 0 es rojo.
Title	string	El título del medidor.
TitleFont	TsgcUITextStyle	La fuente del título. Un Color igual a 0 sigue el tema.
ValueFont	TsgcUITextStyle	La fuente del texto del valor.
BackgroundColor	TsgcUIColor	El color detrás del medidor. 0 usa el fondo del tema.
FaceColor	TsgcUIColor	El color de la esfera. 0 usa el del tema.

Métodos

Método	Descripción

Acknowledge(aIndex)	Reconoce una casilla que no está normal: deja de parpadear.
AcknowledgeAll	Reconoce todas las casillas que no están normales.
TileAt(X, Y)	El índice de la casilla en un punto del control, o -1.
HoverTile	El índice de la casilla bajo el ratón, o -1.

Eventos

OnTileClick(Sender, aIndex) de tipo TsgcGaugeTileClickEvent (unit sgcGauges_Annunciator) se dispara cuando se suelta el botón izquierdo sobre una casilla.

Ejemplo

Tres casillas; un clic reconoce una casilla:

```
uses
  Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Annunciator, sgcGauges_VCL;

type
  TAlarmPanel = class(TComponent)
  public
    Panel: TsgcVCLAnnunciatorGauge;
    procedure TileClick(Sender: TObject; aIndex: Integer);
  end;

procedure TAlarmPanel.TileClick(Sender: TObject; aIndex: Integer);
begin
  // a click acknowledges the tile: it stops flashing and keeps its state
  Panel.Acknowledge(aIndex);
end;

function CreateAlarmPanel(aForm: TForm): TAlarmPanel;
var
  oTile: TsgcGaugeAnnunciatorTile;
begin
  Result := TAlarmPanel.Create(aForm);
  Result.Panel := TsgcVCLAnnunciatorGauge.Create(aForm);
  Result.Panel.Parent := aForm;
  Result.Panel.SetBounds(16, 16, 400, 120);
  Result.Panel.Columns := 3;
  Result.Panel.FlashInterval := 400;
  Result.Panel.Tiles.Add.Caption := 'PUMP 1';
  Result.Panel.Tiles.Add.Caption := 'PUMP 2';
  oTile := Result.Panel.Tiles.Add;
  oTile.Caption := 'OIL PRESS';
  oTile.State := gasAlarm;
  Result.Panel.Tiles[1].State := gasAdvisory;
  Result.Panel.OnTileClick := Result.TileClick;
end;
```

Véase también

- Tipos de medidor
- TsgcCustomGauge

TsgcCustomGauge

Unit: sgcGauges_Model
La clase base de los diez modelos de medidor, un TPersistent sin ventana. Un control host crea el modelo de su tipo y lo devuelve en su propiedad Gauge; un programa sin interfaz de usuario crea el modelo él mismo, como TsgcRadialGauge.Create, y lo libera. Un modelo nuevo tiene una escala de 0 a 100 y un puntero.

Propiedades

Propiedad	Tipo	Descripción
Scales	TsgcGaugeScales	Las escalas. Véase TsgcGaugeScale. Community usa solo la primera.
Pointers	TsgcGaugePointers	Los punteros. Véase TsgcGaugePointer. Community usa solo el primero.
Thresholds	TsgcGaugeThresholds	Los umbrales. Véase TsgcGaugeThreshold.
Animation	TsgcGaugeAnimation	Enabled (por defecto True), Duration en milisegundos (por defecto 300) y Easing (por defecto geCubicOut).
Theme	TsgcUITheme	utLight (por defecto) o utDark.
Title, TitleFont	string, TsgcUITextStyle	El título del medidor y su fuente.
Units	string	El texto dibujado junto al texto del valor. ValueToText no lo añade.
ShowValue	Boolean	Dibuja el texto del valor. Por defecto True.
ValueFormat	string	La máscara FormatFloat del texto del valor. Vacía usa los decimales del paso principal de la escala.
ValueFont	TsgcUITextStyle	La fuente del texto del valor. Un Color igual a 0 sigue el tema.
BackgroundColor, FaceColor	TsgcUIColor	El color detrás del medidor y el color de la esfera. 0 usa el tema.
Value	Double	El valor del puntero 0, igual que SetValue(0, Value).
FormatSettings	TFormatSettings	Pública. Los ajustes del texto del valor y de las etiquetas, con '.' como separador decimal por defecto.
StaticStamp	Integer	Pública, solo lectura. Se incrementa con cada cambio estático; un host la compara para saber si su capa estática en caché está al día.
Clock	TsgcGaugeClockFunc	Pública. Una función que devuelve la hora en milisegundos para la animación. nil usa el reloj del sistema.
Owner	TPersistent	Pública. El control host, o nil para un modelo sin ventana.
Designing	Boolean	Pública. La fijan los hosts en tiempo de diseño, donde nada se anima.

Métodos

Método	Descripción
SetValue(aPointer, aValue, aAnimate = True)	Escribe el valor de un puntero, con una animación cuando aAnimate y Animation.Enabled son True. Seguro para hilos.
BeginUpdate, EndUpdate	Reúnen los cambios entre ambos en un único OnChange.
DisplayValue(aPointer, aNowMs)	El valor dibujado de un puntero en un instante del Clock, durante una animación.
ValueToText(aPointer, aValue)	El texto del valor de un valor, con ValueFormat y OnFormatValue.
IsAnimating(aNowMs), NextFrameMs(aNowMs)	Si un puntero se mueve y cuándo toca el siguiente fotograma temporizado (un parpadeo, un tic del reloj).
ActiveScaleCount, ActivePointerCount, PointerScale(aPointer)	Las escalas y los punteros que dibuja la edición, y la escala de un puntero.
Render, RenderStatic, RenderDynamic	Dibujan las dos capas, o una, en un TsgcUICanvas de aWidth x aHeight píxeles lógicos con aScale.
SaveToPNG, SaveToSVG	Guardan el medidor en un stream o en un archivo con aWidth x aHeight píxeles lógicos, con aScale.

Eventos

Evento	Descripción
OnChange(Sender, aKind)	Un cambio: gckStatic para la capa estática, gckDynamic para un valor o un movimiento de puntero. Se llama en el hilo del cambio.
OnThreshold(Sender, aThreshold, aValue, aRising)	Un puntero cruzó un umbral. En un modelo se llama en el hilo que escribió el valor; los controles lo llaman en el hilo principal.
OnFormatValue(Sender, aPointer, aValue, var aText)	Reemplaza el texto del valor.

Ejemplo

Un medidor sin ventana con dos punteros que cuenta sus cambios:

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Radial;

type
  TGaugeWatcher = class
  public
    StaticChanges, DynamicChanges: Integer;
    procedure GaugeChange(Sender: TObject; aKind: TsgcGaugeChangeKind);
  end;

procedure TGaugeWatcher.GaugeChange(Sender: TObject;
  aKind: TsgcGaugeChangeKind);
begin
  if aKind = gckStatic then
    Inc(StaticChanges)
```

```
  else
    Inc(DynamicChanges);
end;

procedure TwoPointers(aWatcher: TGaugeWatcher);
var
  oGauge: TsgcRadialGauge;
begin
  oGauge := TsgcRadialGauge.Create;
  try
    oGauge.OnChange := aWatcher.GaugeChange;
    oGauge.BeginUpdate;
    try
      // one static OnChange for the whole block
      oGauge.Title := 'Pressure';
      oGauge.Units := 'bar';
      oGauge.Scales[0].Maximum := 10;
      oGauge.Pointers.Add.Kind := gpkMarker;
    finally
      oGauge.EndUpdate;
    end;
    oGauge.SetValue(0, 6.5, False); // dynamic only
    oGauge.SetValue(1, 8, False);
    WriteLn(oGauge.ValueToText(0, oGauge.Value)); // 6.5
    oGauge.SaveToSVG('pressure.svg', 200, 200);
  finally
    oGauge.Free;
  end;
end;
```

Véase también

- Arquitectura
- Salida sin ventana
- Animación

TsgcGaugeScale y TsgcGaugeRange

Unit: sgcGauges_Model
TsgcGaugeScale es un elemento de la colección Scales (TsgcGaugeScales) de un medidor. Una escala asigna valores a posiciones en el arco o en el eje y dibuja su pista, marcas, etiquetas y rangos de color. TsgcGaugeRange es un elemento de la colección Ranges (TsgcGaugeRanges) de una escala. Los tamaños son fracciones del tamaño del medidor.

TsgcGaugeScale

Propiedad	Tipo	Descripción
Minimum, Maximum	Double	El rango de valores. Por defecto 0 y 100. Defínalos antes de los valores, que se limitan al escribirse.
Mapping	TsgcGaugeScaleMapping	gsmLinear (por defecto) o gsmDecibel, donde una posición sigue la amplitud.
Reversed	Boolean	Recorre la escala de Maximum a Minimum. Por defecto False.
Radius	Double	El radio exterior de la escala en una esfera. Por defecto 1.
Offset	Double	La posición de la banda a lo largo del eje de un medidor lineal. Por defecto 0.
Thickness	Double	El ancho de la pista. Por defecto 0.05.
MajorInterval	Double	El paso entre marcas con etiqueta. 0 (por defecto) elige un paso.
MinorCount	Integer	Las marcas menores entre dos marcas principales. Por defecto 4.
TickPlacement	TsgcGaugeTickPlacement	gtpInside (por defecto), gtpOutside o gtpCross.
TickColor	TsgcUIColor	El color de las marcas. 0 usa el tema.
ShowLabels	Boolean	Dibuja las etiquetas de las marcas principales. Por defecto True.
LabelFormat	string	La máscara FormatFloat de las etiquetas. Vacía usa los decimales del paso.
LabelFont	TsgcUITextStyle	La fuente de las etiquetas.
Ranges	TsgcGaugeRanges	Los rangos de color.
Visible	Boolean	Dibuja la escala. Por defecto True.

MajorStep devuelve el paso en uso y ValueToRatio(aValue) la posición de un valor de 0 a 1.

TsgcGaugeRange

Propiedad	Tipo	Descripción

StartValue, EndValue	Double	Los valores que cubre el rango.
Color	TsgcUIColor	El color de la banda.
Thickness	Double	El ancho de la banda. 0 usa el grosor de la escala.
Caption	string	Un nombre para el rango. No se dibuja.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Radial;

procedure ConfigureScales(aGauge: TsgcRadialGauge);
var
  oScale: TsgcGaugeScale;
  oRange: TsgcGaugeRange;
begin
  aGauge.BeginUpdate;
  try
    oScale := aGauge.Scales[0];
    oScale.Minimum := 0;
    oScale.Maximum := 220;
    oScale.MajorInterval := 20;
    oScale.MinorCount := 3;
    oScale.TickPlacement := gtpInside;
    oScale.LabelFormat := '0';
    oRange := oScale.Ranges.Add;
    oRange.StartValue := 180;
    oRange.EndValue := 220;
    oRange.Color := $FFE34948;
    oRange.Thickness := 0.08;
    // an inner scale in miles per hour (Registered: more than one scale)
    oScale := aGauge.Scales.Add;
    oScale.Maximum := 140;
    oScale.Radius := 0.7;
    oScale.TickPlacement := gtpOutside;
    aGauge.Pointers.Add.ScaleIndex := 1;
  finally
    aGauge.EndUpdate;
  end;
end;
```

Véase también

- Escalas
- Rangos
- TsgcCustomGauge

TsgcGaugePointer, TsgcGaugeThreshold y TsgcGaugeAnimation

Unit: sgcGauges_Model
TsgcGaugePointer es un elemento de la colección Pointers (TsgcGaugePointers), TsgcGaugeThreshold un elemento de la colección Thresholds (TsgcGaugeThresholds), y TsgcGaugeAnimation la propiedad Animation de un medidor.

TsgcGaugePointer

Propiedad	Tipo	Descripción
Kind	TsgcGaugePointerKind	gpkNeedle (por defecto), gpkMarker, gpkBar o gpkRange (una banda de Value a EndValue).
ScaleIndex	Integer	La escala que usa el puntero. Por defecto 0.
Value	Double	El valor. Seguro para hilos; al escribirlo se anima.
EndValue	Double	El final de un puntero gpkRange. Seguro para hilos.
Origin	Double	Dónde empieza una barra. Sin definir (NaN) significa el Minimum de la escala. OriginValue devuelve el valor en uso.
Color	TsgcUIColor	El color. 0 usa el tema.
Width, Length, Offset	Double	El tamaño y la posición. Valores por defecto 0.04, 0.9 y 0.
NeedleStyle	TsgcGaugeNeedleStyle	gndTapered (por defecto), gndLine o gndArrow.
MarkerShape	TsgcGaugeMarkerShape	gmsTriangle (por defecto), gmsCircle, gmsDiamond o gmsLine.
CapVisible	Boolean	Dibuja la tapa sobre el pivote de una aguja. Por defecto True.
Visible	Boolean	Dibuja el puntero. Por defecto True.

TsgcGaugeThreshold

Propiedad	Tipo	Descripción
Value	Double	El valor que se vigila.
PointerIndex	Integer	El puntero que se vigila. Por defecto 0.
Direction	TsgcGaugeThresholdDirection	gtdRising (por defecto), gtdFalling o gtdBoth.

Hysteresis	Double	Cuánto debe retroceder el valor antes de que el umbral pueda dispararse de nuevo. Por defecto 0.
Color	TsgcUIColor	El color de la marca del umbral en la escala. 0 (por defecto) no dibuja marca.
ValueColor	TsgcUIColor	El color del texto del valor mientras se cruza el umbral. Se aplica a los umbrales del puntero 0.
Enabled	Boolean	Por defecto True.
Active	Boolean	Pública, solo lectura. True mientras el valor está más allá del umbral.

TsgcGaugeAnimation

Propiedad	Tipo	Descripción
Enabled	Boolean	Anima los movimientos del puntero. Por defecto True.
Duration	Integer	La duración de un movimiento en milisegundos. Por defecto 300.
Easing	TsgcGaugeEasing	geLinear, geQuadOut, geCubicOut (por defecto), geCubicInOut o geBackOut.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Linear;

procedure ConfigurePointers(aGauge: TsgcLinearGauge);
var
  oPointer: TsgcGaugePointer;
  oThreshold: TsgcGaugeThreshold;
begin
  aGauge.BeginUpdate;
  try
    // pointer 0 is the bar of the linear gauge
    aGauge.Pointers[0].Color := $FF2A78D6;
    // a range pointer: a band from Value to EndValue
    oPointer := aGauge.Pointers.Add;
    oPointer.Kind := gpkRange;
    oPointer.Value := 40;
    oPointer.EndValue := 60;
    // a threshold on pointer 0 with a hysteresis of 2
    oThreshold := aGauge.Thresholds.Add;
    oThreshold.Value := 90;
    oThreshold.Direction := gtdBoth;
    oThreshold.Hysteresis := 2;
    oThreshold.Color := $FFE34948;
    oThreshold.ValueColor := $FFE34948;
    aGauge.Animation.Duration := 500;
    aGauge.Animation.Easing := geBackOut;
  finally
    aGauge.EndUpdate;
  end;
end;
```

Véase también

- Punteros
- Umbrales
- Animación

Escalas

Una escala asigna valores a una posición en el medidor: un arco en un medidor radial, una línea en un medidor lineal. La colección Scales de un medidor siempre tiene al menos una escala, de 0 a 100. Un puntero muestra su valor en la escala indicada por su ScaleIndex.

Propiedades de TsgcGaugeScale

Propiedad	Descripción
Minimum, Maximum	El rango de la escala, 0 y 100 por defecto. Un cambio del rango vuelve a limitar los valores de los punteros en la escala.
Mapping	gsmLinear reparte los valores de forma uniforme. gsmDecibel los reparte en una curva de decibelios, como en un VU meter.
Reversed	La escala va de Maximum a Minimum.
Radius, Offset, Thickness	La posición y el ancho de la banda de la escala, como fracciones del tamaño del medidor. Radius es 1 por defecto, el borde exterior del arco, y Thickness es 0.05. Una segunda escala con Radius igual a 0.7 queda dentro de la primera.
MajorInterval, MinorCount	El paso entre dos marcas con etiqueta, y el número de marcas menores entre ellas (4 por defecto). Un MajorInterval igual a 0 elige un paso redondo para unos 5 intervalos. MajorStep devuelve el paso en uso.
TickPlacement, TickColor	Marcas dentro de la banda (gtpInside), fuera de ella (gtpOutside) o a través de ella (gtpCross), y su color (0 = tema).
ShowLabels, LabelFormat, LabelFont	Las etiquetas de las marcas principales. LabelFormat es una máscara FormatFloat; cuando está vacío, las etiquetas tienen los decimales del paso principal. Las etiquetas que se solaparían se reducen.
Ranges	Las bandas de color de la escala. Véase Rangos.
Visible	Dibuja la escala, True por defecto.

ValueToRatio devuelve la posición de un valor en la escala, de 0 al inicio a 1 al final, con el mapeo y Reversed aplicados.

Varias escalas

En la edición Registered, un medidor dibuja todas las escalas de la colección, por ejemplo km/h por fuera y mph por dentro. En Community solo se dibuja la primera escala. Las demás escalas permanecen en la colección, de modo que un formulario hecho con la edición Registered sigue cargándose.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Radial;

procedure SetupDualScale(aGauge: TsgcRadialGauge);
var
  oOuter, oInner: TsgcGaugeScale;
```

```
begin
  aGauge.BeginUpdate;
  try
    // scale 0, the outer one: km/h
    oOuter := aGauge.Scales[0];
    oOuter.Minimum := 0;
    oOuter.Maximum := 240;
    oOuter.MajorInterval := 20;
    oOuter.MinorCount := 3;
    oOuter.TickPlacement := gtpInside;
    oOuter.LabelFormat := '0';
    // scale 1, inside it: mph (drawn in the Registered edition only)
    oInner := aGauge.Scales.Add;
    oInner.Minimum := 0;
    oInner.Maximum := 150;
    oInner.Radius := 0.7;
    oInner.MajorInterval := 25;
    oInner.MinorCount := 4;
    oInner.TickPlacement := gtpOutside;
    oInner.LabelFont.Size := 8;
    // the needle stays on scale 0
    aGauge.Pointers[0].ScaleIndex := 0;
  finally
    aGauge.EndUpdate;
  end;
end;
```

Véase también

- Rangos
- Punteros
- Ediciones

Rangos

Un rango colorea una parte de una escala, como una banda normal verde y una banda crítica roja. Cada escala tiene su propia colección Ranges de elementos TsgcGaugeRange.

Propiedades de TsgcGaugeRange

Propiedad	Descripción
StartValue, End-Value	Los valores en que empieza y termina la banda, en unidades de la escala. Pueden darse en cualquier orden.
Color	El color de la banda. 0 toma un color de la paleta del tema, según el índice del rango.
Thickness	El ancho de la banda como fracción del tamaño del medidor, dibujada desde el borde exterior de la escala. 0 rellena toda la banda de la escala.
Caption	Un nombre para el rango, que se guarda con él. El medidor no lo dibuja.

Rangos y punteros

Un puntero de barra sin color propio (Color 0) toma el color del rango en que está su valor, de modo que la barra de un medidor lineal se pone roja cuando el valor entra en la banda roja. Un umbral también puede colorear el texto del valor, véase Umbrales.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcUICore_Canvas, sgcGauges_Types, sgcGauges_Model, sgcGauges_Linear;

procedure AddBand(aScale: TsgcGaugeScale; aStart, aEnd: Double;
  aColor: TsgcUIColor; const aCaption: string);
var
  oRange: TsgcGaugeRange;
begin
  oRange := aScale.Ranges.Add;
  oRange.StartValue := aStart;
  oRange.EndValue := aEnd;
  oRange.Color := aColor;
  oRange.Caption := aCaption;
  // the band is a fraction of the gauge size; 0 = the full scale band
  oRange.Thickness := 0.03;
end;

procedure SetupTemperature(aGauge: TsgcLinearGauge);
begin
  aGauge.LinearStyle := glsBullet;
  aGauge.Scales[0].Minimum := -20;
  aGauge.Scales[0].Maximum := 120;
  AddBand(aGauge.Scales[0], -20, 60, $FF2EAD6B, 'Normal');
  AddBand(aGauge.Scales[0], 60, 90, $FFF0A030, 'High');
  AddBand(aGauge.Scales[0], 90, 120, $FFE34948, 'Critical');
  // the bullet target line
  aGauge.Target := 75;
```

```
  aGauge.Value := 68;
end;
```

Véase también

- Escalas
- Umbrales
- Temas

Punteros

Un puntero muestra un valor en una escala. La colección Pointers de un medidor siempre tiene al menos un puntero, y Value es un atajo al valor del primero.

Tipos de puntero

Tipo	Descripción
gpkNeedle	Una aguja desde el centro de un medidor radial, cónica, una línea o una flecha (NeedleStyle), con una tapa sobre el pivote (CapVisible). El tipo por defecto.
gpkMarker	Una pequeña forma en la escala: un triángulo, un círculo, un rombo o una línea (MarkerShape). Útil para una consigna o un objetivo.
gpkBar	Una banda que rellena la escala desde Origin hasta el valor.
gpkRange	Una banda de Value a EndValue, por ejemplo una ventana de tolerancia.

Propiedades de TsgcGaugePointer

Propiedad	Descripción
Kind, ScaleIndex	El tipo de puntero y la escala en la que muestra su valor.
Value, EndValue	El valor, y el final de la banda de un puntero de rango. Ambos se limitan a la escala y pueden escribirse desde cualquier hilo.
Origin	Dónde empieza una barra, limitada a la escala. Cuando no se define, la barra empieza en el Minimum de la escala. Un Origin igual a 0 en una escala de -50 a 50 dibuja una barra que crece hacia la izquierda o hacia la derecha desde el centro.
Color	El color del puntero. Con 0, una aguja o un marcador toma un color de la paleta del tema, una barra el color del rango en que está su valor (o el color de acento) y un puntero de rango un color de acento translúcido.
Width, Length, Offset	El tamaño y la posición del puntero, como fracciones del tamaño del medidor (Width 0.04 y Length 0.9 por defecto).
NeedleStyle, MarkerShape, CapVisible, Visible	La forma de una aguja (gndTapered, gndLine, gndArrow) y de un marcador, la tapa de una aguja, y si el puntero se dibuja.

Varios punteros

En la edición Registered, cada puntero de la colección se dibuja, se anima y se evalúa para los umbrales, por ejemplo una aguja para el valor actual y un marcador para la consigna. En Community solo está activo el primer puntero; los demás permanecen en la colección.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Radial;

procedure SetupPointers(aGauge: TsgcRadialGauge);
var
  oSetpoint, oBand: TsgcGaugePointer;
begin
  aGauge.BeginUpdate;
  try
    // pointer 0: the default needle
    aGauge.Pointers[0].NeedleStyle := gndArrow;
    aGauge.Pointers[0].Length := 0.8;
    // pointer 1: a marker for the setpoint (Registered edition)
    oSetpoint := aGauge.Pointers.Add;
    oSetpoint.Kind := gpkMarker;
    oSetpoint.MarkerShape := gmsTriangle;
    oSetpoint.Color := $FF2EAD6B;
    // pointer 2: a band from Value to EndValue (Registered edition)
    oBand := aGauge.Pointers.Add;
    oBand.Kind := gpkRange;
    oBand.Width := 0.06;
  finally
    aGauge.EndUpdate;
  end;
  aGauge.Pointers[0].Value := 64;
  aGauge.Pointers[1].Value := 70;
  aGauge.Pointers[2].Value := 55;
  aGauge.Pointers[2].EndValue := 85;
end;

procedure SetupDeviation(aGauge: TsgcRadialGauge);
begin
  // a bar that grows from 0, left or right, on a -50..50 scale
  aGauge.Scales[0].Minimum := -50;
  aGauge.Scales[0].Maximum := 50;
  aGauge.Pointers[0].Kind := gpkBar;
  aGauge.Pointers[0].Origin := 0;
  aGauge.Value := -18;
end;
```

Véase también

- Escalas
- Animación
- Umbrales

Umbrales

Un umbral vigila el valor de un puntero y dispara OnThreshold cuando el valor lo cruza. Úselo para alarmas, advertencias o un cambio de color del texto del valor.

Cómo funciona un umbral

- Un umbral se evalúa cuando se escribe un valor, sobre el nuevo valor de destino, nunca sobre los valores dibujados durante una animación.
- gtdRising y gtdBoth están activos en Value o por encima y dejan de estarlo por debajo de Value menos Hysteresis. gtdFalling está activo en Value o por debajo y deja de estarlo por encima de Value más Hysteresis. La histéresis evita que un valor que oscila alrededor del límite se dispare una y otra vez.
- Con gtdRising y gtdFalling, OnThreshold se dispara cuando el umbral pasa a estar activo. Con gtdBoth se dispara en los dos cruces, y aRising indica la dirección. Para una alarma que también necesita un evento de fin, use gtdBoth con una histéresis.
- Un cambio de configuración, como un nuevo Value del umbral, actualiza Active sin evento.

Propiedades de TsgcGaugeThreshold

Propiedad	Descripción
Value, Hysteresis, Direction	El límite, la banda muerta y la dirección (gtdRising por defecto).
PointerIndex	El puntero que vigila el umbral, 0 por defecto.
Color	Cuando no es 0, se dibuja en la escala, en Value, una marca de este color.
ValueColor	Cuando no es 0, el texto del valor toma este color mientras el umbral está activo (umbrales del primer puntero).
Enabled, Active	Desactiva el umbral e indica si está activo ahora (solo lectura).

OnThreshold

`procedure(Sender: TObject; aThreshold: TsgcGaugeThreshold; aValue: Double; aRising: Boolean)`. Los controles VCL y FireMonkey lo llaman en el hilo principal, en el orden de los cruces, sea cual sea el hilo que escribió el valor. Un modelo sin host lo llama en el hilo que escribió el valor.

Ejemplo

```
uses
  SysUtils, Classes, Controls, Forms, StdCtrls,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

type
  TTankAlarms = class(TComponent)
  public
    Level: TsgcVCLLinearGauge;
    Status: TLabel;
    procedure LevelThreshold(Sender: TObject; aThreshold: TsgcGaugeThreshold;
      aValue: Double; aRising: Boolean);
  end;

procedure TTankAlarms.LevelThreshold(Sender: TObject;
```

```
  aThreshold: TsgcGaugeThreshold; aValue: Double; aRising: Boolean);
begin
  case aThreshold.Index of
    0:
      if aRising then
        Status.Caption := Format('High level: %.1f %%', [aValue])
      else
        Status.Caption := 'Level back to normal';
    1:
      Status.Caption := Format('Low level: %.1f %%', [aValue]);
  end;
end;

procedure SetupTankAlarms(aAlarms: TTankAlarms);
var
  oHigh, oLow: TsgcGaugeThreshold;
begin
  // threshold 0: active at 90 %, clears below 85 %, an event both ways
  oHigh := aAlarms.Level.Thresholds.Add;
  oHigh.Value := 90;
  oHigh.Hysteresis := 5;
  oHigh.Direction := gtdBoth;
  oHigh.Color := $FFE34948;
  oHigh.ValueColor := $FFE34948;
  // threshold 1: active at 10 % or below, an event on entering only
  oLow := aAlarms.Level.Thresholds.Add;
  oLow.Value := 10;
  oLow.Hysteresis := 5;
  oLow.Direction := gtdFalling;
  oLow.Color := $FFF0A030;
  aAlarms.Level.OnThreshold := aAlarms.LevelThreshold;
end;
```

Véase también

- Rangos
- Punteros
- Inicio rápido VCL

Animación

Cuando un valor cambia, el puntero se mueve del valor que muestra al nuevo en un tiempo breve, con una curva de suavizado. El valor en sí, los umbrales y los eventos cambian de golpe; solo se mueve el dibujo.

Propiedades de TsgcGaugeAnimation

Propiedad	Descripción
Enabled	Anima los cambios de valor, True por defecto.
Duration	El tiempo de un movimiento en milisegundos, 300 por defecto. 0 desactiva la animación.
Easing	La curva del movimiento: geLinear, geQuadOut, geCubicOut (la predeterminada), geCubicInOut o geBackOut, que se pasa un poco al final.

Cuándo un valor no se anima

- En tiempo de diseño, con Enabled en False o con Duration igual a 0.
- Cuando el valor se escribe con `Gauge.SetValue(aPointer, aValue, False)`.
- Cuando el valor se carga desde un archivo de formulario, mientras se actualiza la colección Pointers.
- En un renderizado sin ventana, que siempre dibuja el valor de destino.

Un nuevo valor que llega durante un movimiento parte del valor dibujado en ese instante, de modo que el puntero nunca da saltos.

Fotogramas

Los controles VCL y FireMonkey dibujan los fotogramas de un movimiento solo en la capa dinámica, y como máximo MaxFPS veces por segundo (60 por defecto). Cuando nada se mueve, un medidor no dibuja ningún fotograma. La hora del medidor procede de la función Clock del modelo; DisplayValue devuelve el valor dibujado en una hora dada e IsAnimating indica si un puntero sigue moviéndose.

Ejemplo

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

procedure SetupAnimation(aGauge: TsgcVCLRadialGauge);
begin
  // half a second, with a small overshoot at the end
  aGauge.Animation.Enabled := True;
  aGauge.Animation.Duration := 500;
  aGauge.Animation.Easing := geBackOut;
  // at most 30 frames per second while a pointer moves
  aGauge.MaxFPS := 30;
end;

procedure ShowReading(aGauge: TsgcVCLRadialGauge; aValue: Double);
begin
  // animated, as the Animation settings say
  aGauge.Value := aValue;
end;

procedure ResetReading(aGauge: TsgcVCLRadialGauge);
```

```
begin
  // no animation: the needle jumps to 0
  aGauge.Gauge.SetValue(0, 0, False);
end;

procedure ShowMinMax(aGauge: TsgcVCLRadialGauge; aMin, aMax: Double);
begin
  // one change and one repaint for both pointers
  aGauge.Gauge.BeginUpdate;
  try
    aGauge.Gauge.SetValue(1, aMin);
    aGauge.Gauge.SetValue(2, aMax);
  finally
    aGauge.Gauge.EndUpdate;
  end;
end;
```

Véase también

- Punteros
- Arquitectura
- Salida sin ventana

Temas

Un medidor toma sus colores del tema de sgcUICore definido en **Theme**: utLight (el predeterminado) o utDark. Toda propiedad de color de un medidor es 0 por defecto, lo que significa el color del tema, de modo que un medidor sin colores propios sigue por completo el tema.

Colores

Propiedad	Con 0
BackgroundColor	El fondo del tema.
FaceColor	El color de la esfera del tema, detrás de la escala de una esfera.
Scales[i].TickColor	El color de texto atenuado del tema.
Ranges[i].Color, Pointers[i].Color	Colores de la paleta del tema, véase Rangos y Punteros.
TitleFont.Color, ValueFont.Color, LabelFont.Color	El color de texto del tema.

Los colores se escriben como $AARRGGBB, por ejemplo $FFE34948 para un rojo opaco.

Texto

- TitleFont, ValueFont y el LabelFont de cada escala son valores TsgcUITextStyle con Size, Bold y Color. Los tamaños son relativos al medidor: el texto crece y disminuye con el control. El texto del valor se dibuja al doble del tamaño de ValueFont.
- ValueFormat es una máscara FormatFloat para el texto del valor. Cuando está vacía, el valor tiene los decimales de la escala. Units se dibuja junto al valor.
- Los números usan el FormatSettings del modelo, con '.' como separador decimal por defecto, de modo que un medidor tiene el mismo aspecto en cualquier configuración regional hasta que lo cambie.
- OnFormatValue permite reemplazar el texto del valor de cualquier puntero.

Ejemplo

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcUICore_Theme, sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

type
  TGaugeStyler = class(TComponent)
  public
    procedure FormatPower(Sender: TObject; aPointer: Integer; aValue: Double;
      var aText: string);
  end;

procedure TGaugeStyler.FormatPower(Sender: TObject; aPointer: Integer;
  aValue: Double; var aText: string);
begin
  if aValue >= 1000 then
    aText := FormatFloat('0.00', aValue / 1000) + ' MW';
end;

procedure StyleGauge(aGauge: TsgcVCLRadialGauge; aStyler: TGaugeStyler);
var
```

```
  vSettings: TFormatSettings;
begin
  aGauge.Theme := utDark;
  // 0 means the theme colour, so only these two change
  aGauge.FaceColor := $FF1E2A38;
  aGauge.Pointers[0].Color := $FFF0A030;
  aGauge.Scales[0].TickColor := $FF8899AA;
  aGauge.TitleFont.Size := 11;
  aGauge.TitleFont.Bold := True;
  aGauge.ValueFormat := '0.0';
  // labels and values with a decimal comma
  vSettings := aGauge.Gauge.FormatSettings;
  vSettings.DecimalSeparator := ',';
  aGauge.Gauge.FormatSettings := vSettings;
  aGauge.OnFormatValue := aStyler.FormatPower;
end;
```

Véase también

- Rangos
- Inicio rápido FMX
- Salida sin ventana

Salida sin ventana

Todo medidor se renderiza sin ventana. Las clases de modelo dibujan en un canvas de sgcUICore y guardan archivos PNG y SVG, de modo que un servidor, un informe o un programa de consola en Windows o Linux64 puede producir las mismas imágenes que los controles.

Métodos de un modelo

Método	Descripción
SaveToPNG(aStream or aFileName, aWidth, aHeight, aScale = 1)	Una imagen PNG de aWidth x aHeight píxeles lógicos, multiplicada por aScale. Una escala de 2 da una imagen del doble de tamaño para una pantalla de DPI alto o una impresión.
SaveToSVG(aStream or aFileName, aWidth, aHeight, aScale = 1)	Un documento SVG del mismo dibujo. Los números del SVG nunca dependen de la configuración regional.
Render(aCanvas, aWidth, aHeight, aScale = 1, aNowMs = -1)	Dibuja las dos capas en un TsgcUICanvas. Con aNowMs por debajo de 0, los punteros se dibujan en sus valores de destino.
RenderStatic, RenderDynamic	Dibujan solo la capa estática o solo la capa dinámica.

Controles

Los controles VCL y FireMonkey tienen SaveToPNG(aStream or aFileName, aScale = 1) y SaveToSVG con los mismos argumentos. Guardan el medidor con el tamaño de cliente del control, de modo que una captura de un panel no necesita captura de ventana.

La misma salida en todas partes

La salida sin ventana es determinista: el mismo medidor da archivos PNG y SVG idénticos byte a byte en Win32, Win64 y Linux64. Las units del modelo no usan VCL ni FireMonkey y, en Linux64, compilan desde Delphi 10.2.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_Linear;

function TankLevelSVG(aLevel: Double): string;
var
  oGauge: TsgcLinearGauge;
  oStream: TStringStream;
begin
  oGauge := TsgcLinearGauge.Create;
  try
    oGauge.Orientation := gloVertical;
    oGauge.Units := '%';
    oGauge.Value := aLevel;
    // Delphi 2009 and later; the SVG is UTF-8 text
    oStream := TStringStream.Create('', TEncoding.UTF8);
    try
      // 80 x 240 logical pixels
      oGauge.SaveToSVG(oStream, 80, 240);
      Result := oStream.DataString;
```

```
    finally
      oStream.Free;
    end;
  finally
    oGauge.Free;
  end;
end;

procedure SaveTankLevelPNG(aStream: TStream; aLevel: Double);
var
  oGauge: TsgcLinearGauge;
begin
  oGauge := TsgcLinearGauge.Create;
  try
    oGauge.Orientation := gloVertical;
    oGauge.Value := aLevel;
    // twice the size, for a high DPI page or a report
    oGauge.SaveToPNG(aStream, 80, 240, 2);
  finally
    oGauge.Free;
  end;
end;
```

Véase también

- Inicio rápido sin ventana
- Plataformas
- Temas

Ediciones

sgcGauges se compila a partir del mismo código fuente para todas las ediciones.

Edición	Incluye
Registered	Todo: los diez tipos de medidor para VCL y FireMonkey, varias escalas y punteros por medidor, las pantallas de 14 segmentos y de matriz de puntos, y el puente sgcGaugesWS hacia sgcWebSockets.
Trial	Las mismas características que Registered. Una compilación hecha con SGC_TRIAL muestra un mensaje de que es una versión de prueba cuando se crea un componente después de pasada la ventana de lanzamiento indicada en la compilación. No cambia nada más.
Community	Seis tipos de medidor para VCL y FireMonkey sin las características avanzadas que se enumeran más abajo, y sin aviso de prueba.

Qué incluye Community

El medidor radial (todos los estilos), el medidor lineal (barra, termómetro y bullet), la pantalla digital (7 segmentos), el LED, la barra de LEDs y el anillo de progreso, con rangos, umbrales, animación, los temas claro y oscuro, los controles VCL y FireMonkey y la salida PNG y SVG sin ventana. El objetivo bullet del medidor lineal y el pico de la barra de LEDs son propiedades, no punteros adicionales, de modo que funcionan en Community.

Las características avanzadas

La define SGC_GAUGES_ADVANCED de sgcVer.inc las activa. Está desactivada en la edición Community.

- **Varias escalas y punteros**: las escalas y los punteros posteriores al primero se conservan en sus colecciones, de modo que un formulario hecho con la edición Registered sigue cargándose, pero Community no los dibuja, no los anima ni evalúa umbrales para ellos.
- **Cuatro tipos de medidor**: el VU meter, el reloj, la brújula y el anunciador, sus units (sgcGauges_VUMeter, sgcGauges_Clock, sgcGauges_Compass, sgcGauges_Annunciator), sus ocho controles y su registro.
- **Pantallas digitales**: el dibujo de 14 segmentos y de matriz de puntos. gdt14Segment y gdtDotMatrix permanecen en la enumeración y se dibujan como 7 segmentos.
- **El puente**: los paquetes sgcGaugesWS no se compilan ni se instalan en Community.

Código para todas las ediciones

El código que usa un tipo avanzado puede comprobar la define, de modo que la misma unit compila en las dos ediciones.

```
{$I sgcVer.inc}

uses
  SysUtils, Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

function CreateHeadingGauge(aForm: TForm): TsgcVCLCustomGauge;
begin
{$IFDEF SGC_GAUGES_ADVANCED}
  // Registered and Trial: the compass
  Result := TsgcVCLCompassGauge.Create(aForm);
  TsgcVCLCompassGauge(Result).Heading := 270;
{$ELSE}
  // Community: a full circle radial gauge from 0 to 360
  Result := TsgcVCLRadialGauge.Create(aForm);
```

```
  TsgcVCLRadialGauge(Result).Scales[0].Maximum := 360;
  TsgcVCLRadialGauge(Result).Scales[0].MajorInterval := 45;
  TsgcVCLRadialGauge(Result).Value := 270;
{$ENDIF}
  Result.Parent := aForm;
  Result.SetBounds(16, 16, 200, 200);
end;
```

Véase también

- Tipos de medidor
- Instalación
- Solución de problemas

Descripción general del puente

sgcGaugesWS es el puente entre sgcGauges y sgcWebSockets. Sus fuentes leen los mensajes JSON que recibe un cliente sgcWebSockets y escriben los valores en los medidores, sin código en los eventos de mensaje. El puente forma parte de la edición Registered y necesita sgcWebSockets; tiene sus propios paquetes, sgcGaugesWS y dclsgcGaugesWS, y sus componentes están en la página **SGC Gauges** de la paleta.

Componente	Unit	Cliente
TsgcGaugeFeedSource_MQTT	sgcGaugesWS_Source_MQTT	TsgcWSPClient_MQTT
TsgcGaugeFeedSource_WebSocket	sgcGaugesWS_Source_WebSocket	TsgcWebSocketClient
TsgcGaugeFeedSource_SSE	sgcGaugesWS_Source_SSE	TsgcHTTP1Client

Cómo funciona una fuente

Mientras una fuente está Active y su Client está asignado, en tiempo de ejecución, instala su propio manejador de mensajes en el cliente y conserva el manejador que había antes, de modo que el manejador de la aplicación sigue disparándose después de la fuente. Poner Active en False, limpiar Client o liberar la fuente restaura el manejador anterior. La fuente no conecta, no se suscribe ni reconecta: la aplicación lo hace con el cliente.

Enlaces

Cada elemento de Bindings (TsgcGaugeFeedBinding, unit sgcGaugesWS_Source) escribe un puntero de un medidor. Un mensaje se analiza una sola vez como JSON y lo leen todos los enlaces activos.

Propiedad	Tipo	Descripción
Gauge	TComponent	Un control de medidor VCL o FireMonkey. Se rechazan otros componentes.
GaugeModel	TsgcCustomGauge	Pública. Un modelo sin ventana; tiene prioridad sobre Gauge. Límpiela antes de liberar el modelo.
PointerIndex	Integer	El puntero escrito. Por defecto 0.
ValuePath	string	La ruta JSON del valor, como price o data.items[0].v. Vacía lee el mensaje completo.
FilterPath	string	La ruta de un campo que selecciona los mensajes. El enlace solo lee un mensaje cuando la ruta existe.
FilterValue	string	Si no está vacío, el texto de FilterPath debe ser igual a él, como ticker.
Topic	string	Solo MQTT: un filtro de tema con + y #. Vacío lee todas las publicaciones.
EventName	string	Solo SSE: el tipo de evento. Vacío lee todos los eventos.
Active	Boolean	False hace que el enlace ignore todos los mensajes. Por defecto True.

Se aceptan un número JSON, una cadena numérica con '.' como separador decimal y un booleano (true es 1, false es 0). Un mensaje sin la ruta de un enlace es omitido por ese enlace, de modo que las actualizaciones parciales funcionan. Un JSON no válido, un valor de otro tipo o un índice de puntero incorrecto van a OnError.

Propiedades y eventos de la fuente

Miembro	Descripción
Client	El cliente sgcWebSockets que lee la fuente.
Active	False hace que la fuente ignore todos los mensajes. Por defecto True.
Bindings	Los enlaces.
OnValue(Sender, aBinding, aValue)	Un enlace escribió un valor.
OnError(Sender, aError)	No se pudo aplicar un mensaje.

Hilos

Los valores se escriben con el SetValue seguro para hilos del medidor, en el hilo en que llega el mensaje, y los controles de medidor se repintan y llaman a OnThreshold en el hilo principal. Las escrituras de un mensaje en un medidor se convierten en un único cambio. OnValue y OnError se ejecutan en el hilo de llegada bajo el bloqueo de la fuente: use queue (TThread.Queue) en lugar de synchronize desde ellos antes de tocar un control. Cambie Bindings desde el hilo principal.

Compilación

Mantenga separadas las carpetas de código fuente de sgcGauges y de sgcWebSockets en la ruta de búsqueda. Cada carpeta tiene su propio sgcVer.inc, y una unit lee el de su propia carpeta.

Ejemplo

Cada fuente tiene un método público que aplica un mensaje como si el cliente lo hubiera recibido: ProcessPublish (MQTT), ProcessMessage (WebSocket) y ProcessChunk (SSE). Devuelven el número de valores escritos. Esto alimenta un modelo sin ventana, sin cliente:

```
uses
  SysUtils, Classes,
  sgcGauges_Model, sgcGauges_Radial,
  sgcGaugesWS_Source, sgcGaugesWS_Source_WebSocket;

procedure FeedWithoutClient;
var
  oGauge: TsgcRadialGauge;
  oFeed: TsgcGaugeFeedSource_WebSocket;
  oBinding: TsgcGaugeFeedBinding;
begin
  oGauge := TsgcRadialGauge.Create;
  oFeed := TsgcGaugeFeedSource_WebSocket.Create(nil);
  try
    oGauge.Scales[0].Maximum := 200;
    oBinding := oFeed.Bindings.Add;
    oBinding.GaugeModel := oGauge; // a headless model
    oBinding.ValuePath := 'data.price';
    oBinding.FilterPath := 'type';
    oBinding.FilterValue := 'ticker';
    // the messages a client would deliver
    oFeed.ProcessMessage('{"type":"ticker","data":{"price":101.5}}');
```

```
    oFeed.ProcessMessage('{"type":"trade","data":{"price":99}}'); // filtered
    WriteLn(FloatToStr(oGauge.Value)); // 101.5
  finally
    oFeed.Free; // free the source before the model it writes to
    oGauge.Free;
  end;
end;
```

Véase también

- Fuente MQTT
- Fuente WebSocket
- Fuente de eventos enviados por el servidor
- TsgcCustomGauge

Fuente MQTT

Clase: TsgcGaugeFeedSource_MQTT (unit sgcGaugesWS_Source_MQTT)
Alimenta medidores con las publicaciones que recibe un TsgcWSPClient_MQTT. Client es el cliente MQTT. La fuente instala su manejador OnMQTTPublish y conserva el anterior, y el gancho sigue funcionando tras las reconexiones.
El Topic de cada enlace es un filtro de tema MQTT: + coincide con un nivel y # con este nivel y todos los inferiores. Un enlace lee solo las publicaciones cuyo tema coincide con él, y un Topic vacío lee todas las publicaciones. La fuente no se suscribe, así que la aplicación se suscribe en el cliente, con un filtro que cubra los temas de los enlaces. Cada payload se lee como JSON.
ProcessPublish(aTopic, aText) aplica una publicación sin cliente y devuelve el número de valores escritos. La función sgcGaugeMQTTTopicMatch(aFilter, aTopic) de la misma unit comprueba un tema contra un filtro.

Ejemplo

Dos medidores radiales alimentados por los temas sensors/<id>/temperature y sensors/<id>/humidity, con payloads como {"value": 21.5}:

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcWebSocket, sgcWebSocket_Protocols,
  sgcGauges_VCL, sgcGaugesWS_Source, sgcGaugesWS_Source_MQTT;

type
  TSensorFeed = class(TComponent)
  public
    WSClient: TsgcWebSocketClient;
    MQTT: TsgcWSPClient_MQTT;
    Feed: TsgcGaugeFeedSource_MQTT;
    procedure FeedError(Sender: TObject; const aError: string);
  end;

procedure TSensorFeed.FeedError(Sender: TObject; const aError: string);
begin
  // the thread of the client: queue before touching a control
end;

function CreateSensorFeed(aForm: TForm; aTemperature,
  aHumidity: TsgcVCLRadialGauge): TSensorFeed;
var
  oBinding: TsgcGaugeFeedBinding;
begin
  Result := TSensorFeed.Create(aForm);
  Result.WSClient := TsgcWebSocketClient.Create(Result);
  Result.WSClient.Host := 'broker.example.com';
  Result.WSClient.Port := 1883;
  Result.MQTT := TsgcWSPClient_MQTT.Create(Result);
  Result.MQTT.Client := Result.WSClient;

  Result.Feed := TsgcGaugeFeedSource_MQTT.Create(Result);
  Result.Feed.OnError := Result.FeedError;
  oBinding := Result.Feed.Bindings.Add;
  oBinding.Topic := 'sensors/+/temperature';
  oBinding.ValuePath := 'value';
  oBinding.Gauge := aTemperature;
  oBinding := Result.Feed.Bindings.Add;
  oBinding.Topic := 'sensors/+/humidity';
  oBinding.ValuePath := 'value';
  oBinding.Gauge := aHumidity;
  Result.Feed.Client := Result.MQTT;

  Result.WSClient.Active := True;
```

```
  // the source does not subscribe: the application does
  Result.MQTT.Subscribe('sensors/#');
end;
```

Véase también

- Descripción general del puente
- Fuente WebSocket
- Fuente de eventos enviados por el servidor

Fuente WebSocket

Clase: TsgcGaugeFeedSource_WebSocket (unit sgcGaugesWS_Source_WebSocket)
Alimenta medidores con los mensajes de texto que recibe un TsgcWebSocketClient. Client es el cliente WebSocket. La fuente instala su manejador OnMessage y conserva el anterior, y el gancho sigue funcionando tras las reconexiones, como las del WatchDog. Cada mensaje de texto se lee como JSON con ValuePath, FilterPath y FilterValue de los enlaces, en el hilo desde el que notifica el cliente, de modo que cualquier modo NotifyEvents del cliente es seguro.
FilterPath y FilterValue seleccionan los mensajes de un flujo que transporta varios tipos, como un canal de ticker y otro de operaciones en una sola conexión. ProcessMessage(aText) aplica un mensaje sin cliente y devuelve el número de valores escritos.

Ejemplo

Un display de precio y una barra de variación alimentados por los mensajes ticker de un flujo:

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcWebSocket,
  sgcGauges_VCL, sgcGaugesWS_Source, sgcGaugesWS_Source_WebSocket;

type
  TTickerFeed = class(TComponent)
  public
    WSClient: TsgcWebSocketClient;
    Feed: TsgcGaugeFeedSource_WebSocket;
    procedure FeedValue(Sender: TObject; aBinding: TsgcGaugeFeedBinding;
      aValue: Double);
  end;

procedure TTickerFeed.FeedValue(Sender: TObject;
  aBinding: TsgcGaugeFeedBinding; aValue: Double);
begin
  // the thread of the client, under the source lock: keep it short
end;

function CreateTickerFeed(aForm: TForm; aPrice: TsgcVCLDigitalGauge;
  aChange: TsgcVCLLinearGauge): TTickerFeed;
var
  oBinding: TsgcGaugeFeedBinding;
begin
  Result := TTickerFeed.Create(aForm);
  Result.WSClient := TsgcWebSocketClient.Create(Result);
  Result.WSClient.URL := 'wss://feed.example.com/ticker';

  Result.Feed := TsgcGaugeFeedSource_WebSocket.Create(Result);
  Result.Feed.Client := Result.WSClient;
  Result.Feed.OnValue := Result.FeedValue;
  // {"type":"ticker","symbol":"ABC","last":"12.50","change":0.8}
  oBinding := Result.Feed.Bindings.Add;
  oBinding.FilterPath := 'type';
  oBinding.FilterValue := 'ticker';
  oBinding.ValuePath := 'last';
  oBinding.Gauge := aPrice;
  oBinding := Result.Feed.Bindings.Add;
  oBinding.FilterPath := 'type';
  oBinding.FilterValue := 'ticker';
  oBinding.ValuePath := 'change';
  oBinding.Gauge := aChange;
```

```
  Result.WSClient.Active := True;
end;
```

Véase también

- Descripción general del puente
- Fuente MQTT
- Fuente de eventos enviados por el servidor

Fuente de eventos enviados por el servidor

Clase: TsgcGaugeFeedSource_SSE (unit sgcGaugesWS_Source_SSE)
Alimenta medidores con los eventos enviados por el servidor de un TsgcHTTP1Client. Client es el cliente HTTP. La fuente instala su manejador OnSSEMessage y conserva el anterior. La aplicación inicia el flujo con Client.GetSSE. La fuente divide los chunks del flujo en eventos: las líneas data: se unen con un salto de línea, una línea event: fija el tipo de evento y una línea vacía termina el evento. Se aceptan finales de línea LF, CRLF y CR, y un evento puede dividirse entre chunks. El EventName de cada enlace selecciona el tipo de evento que lee. Un evento sin línea event: tiene el tipo message, y un EventName vacío lee todos los eventos. Los datos de cada evento se leen como JSON.

Método	Descripción
ResetStream	Descarta un evento que un flujo dejó a medio leer. Llámelo antes de un nuevo flujo tras una reconexión.
ProcessChunk(aText)	Aplica un chunk de un flujo sin cliente y devuelve el número de valores escritos.

Ejemplo

Un anillo de progreso y una barra de LEDs alimentados por los eventos cpu y queue de un flujo de métricas:

```
uses
  SysUtils, Classes, Controls, Forms,
  sgcHTTP,
  sgcGauges_VCL, sgcGaugesWS_Source, sgcGaugesWS_Source_SSE;

type
  TMetricsFeed = class(TComponent)
  public
    HTTP: TsgcHTTP1Client;
    Feed: TsgcGaugeFeedSource_SSE;
  end;

function CreateMetricsFeed(aForm: TForm; aCPU: TsgcVCLProgressRingGauge;
  aQueue: TsgcVCLLEDBarGauge): TMetricsFeed;
var
  oBinding: TsgcGaugeFeedBinding;
begin
  Result := TMetricsFeed.Create(aForm);
  Result.HTTP := TsgcHTTP1Client.Create(Result);
  Result.Feed := TsgcGaugeFeedSource_SSE.Create(Result);
  Result.Feed.Client := Result.HTTP;
  // event: cpu       data: {"load":42.5}
  oBinding := Result.Feed.Bindings.Add;
  oBinding.EventName := 'cpu';
  oBinding.ValuePath := 'load';
  oBinding.Gauge := aCPU;
  // event: queue     data: {"pending":17}
  oBinding := Result.Feed.Bindings.Add;
  oBinding.EventName := 'queue';
  oBinding.ValuePath := 'pending';
  oBinding.Gauge := aQueue;
end;

procedure StartMetrics(aFeed: TMetricsFeed);
begin
  aFeed.HTTP.GetSSE('https://metrics.example.com/stream');
end;
```

```
procedure RestartMetrics(aFeed: TMetricsFeed);
begin
  // a new stream: drop an event that the old stream left half read
  aFeed.Feed.ResetStream;
  aFeed.HTTP.GetSSE('https://metrics.example.com/stream');
end;
```

Véase también

- Descripción general del puente
- Fuente MQTT
- Fuente WebSocket

Descripción general de .NET

sgcGauges .NET es una versión C# gestionada de la biblioteca sgcGauges de Delphi, no un envoltorio alrededor de ella. Las clases viven en el espacio de nombres **esegece.sgcGauges** y conservan los nombres de Delphi: **TsgcCustomGauge**, los diez modelos de medidor como **TsgcRadialGauge** y **TsgcLEDBarGauge**, **TsgcGaugeScale**, **TsgcGaugeRange**, **TsgcGaugePointer**, **TsgcGaugeThreshold**, **TsgcGaugeAnimation** y las enumeraciones con sus prefijos de medidor (**grsHalf**, **gtdBoth**, **glsBullet**). Los controles siguen el patrón de nombres de Delphi en todos los hosts: **TsgcVCLRadialGauge** es **TsgcWinFormsRadialGauge**, **TsgcWpfRadialGauge** y **TsgcAvaloniaRadialGauge**. Los medidores se dibujan con la versión .NET de **sgcUICore**, el mismo canvas, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe.
Como en Delphi, cada control posee un modelo independiente del framework, expuesto como **Gauge**. El modelo guarda las escalas, los rangos, los punteros, los umbrales y la animación y se dibuja en cualquier canvas de sgcUICore, así que el mismo código de modelo funciona en un formulario WinForms, en una ventana WPF o Avalonia y en un programa de consola sin interfaz, véase Renderizado sin ventana en .NET.

Paquetes

La versión se divide en seis paquetes NuGet, de modo que un proyecto solo incorpora lo que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcGauges	Los diez modelos de medidor, las clases de escala, rango, puntero, umbral y animación, el painter y la salida SVG (**SaveToSVG**). Sin dependencias nativas. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGauges.Rendering	La salida PNG sin ventana: los métodos de extensión **SaveToPNG** y **RenderToRaster** de **sgcGauges_Raster**, dibujados en el canvas SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGauges.WinForms	Los diez controles **TsgcWinForms<Type>Gauge** sobre **TsgcWinFormsCustomGauge**.	net472, net8.0-windows, net9.0-windows
esegece.sgcGauges.Wpf	Los diez controles **TsgcWpf<Type>Gauge** sobre **TsgcWpfCustomGauge**.	net472, net8.0-windows, net9.0-windows
esegece.sgcGauges.Avalonia	Los diez controles **TsgcAvalonia<Type>Gauge** sobre **TsgcAvaloniaCustomGauge**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0
esegece.sgcGauges.sgcWebSockets	El puente hacia sgcWebSockets: **TsgcGaugeFeedSource_WebSocket**, **TsgcGaugeFeedSource_MQTT** y **TsgcGaugeFeedSource_SSE**. Solo en la edición Registered.	netstandard2.0, net8.0, net9.0

Una línea `PackageReference` trae todo lo que necesita un host: los paquetes WinForms y Avalonia dependen de esegece.sgcGauges.Rendering, que depende de esegece.sgcGauges; el paquete WPF depende de esegece.sgcGauges y dibuja mediante esegece.sgcUICore.Wpf. Véase Instalación desde NuGet.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcGauges, .Rendering, .WinForms, .Wpf, .Avalonia y .sgcWebSockets	Todo: los diez tipos de medidor en los tres hosts y sin ventana, varias escalas y punteros por medidor, los 14 displays de segmentos y de matriz de puntos, y el puente sgcWebSockets.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcGauges.WinForms.Community. No hay paquete de puente Community.	Los medidores radial, lineal, digital, LED, barra de LED y anillo de progreso. El VU meter, el reloj, la brújula y el anunciador no están en los ensamblados Community; solo la primera escala y el primer puntero de un medidor se dibujan, se animan y se comprueban frente a los umbrales; **gdt14Segment** y **gdtDotMatrix** se dibujan como 7 segmentos.

Es la misma división que las ediciones de Delphi, véase Ediciones.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (controles TsgcVCL)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	VCL y FMX	Windows, .NET Framework 4.7.2 y .NET 8 y 9. Los controles se pueden declarar en XAML.
Avalonia	FMX (controles TsgcFMX)	Windows, Linux y macOS, .NET 8 y 9.
Sin ventana	Los modelos de medidor	Cualquier plataforma de los frameworks de destino: SVG con esegece.sgcGauges, PNG con esegece.sgcGauges.Rendering.

Todo control publica las propiedades de su equivalente Delphi con los mismos nombres, además de **Gauge**, **Theme**, **OnThreshold**, **MaxFPS**, **SaveToPNG** y **SaveToSVG** en la clase base. Véase Clases de .NET para los nombres de las clases y Diferencias de .NET respecto a Delphi para lo que cambia respecto a Delphi.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi

- Enlace de datos en .NET
- Renderizado sin ventana en .NET
- Puente sgcWebSockets en .NET
- Demos de .NET
- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, los paquetes del núcleo y del renderizado llegan con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcGauges.WinForms
dotnet add package esegece.sgcGauges.Wpf
dotnet add package esegece.sgcGauges.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcGauges.WinForms.Community
dotnet add package esegece.sgcGauges.Wpf.Community
dotnet add package esegece.sgcGauges.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcGauges.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los Community, nunca los dos: contienen los mismos tipos en el mismo espacio de nombres.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:sgc="clr-namespace:esegece.sgcGauges;assembly=esegece.sgcGauges.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. Asigne el espacio de nombres en XAML con `xmlns:sgc="using:esegece.sgcGauges"`.
- Sin ventana: esegece.sgcGauges solo basta para los modelos y el SVG. Añada esegece.sgcGauges.Rendering para PNG, que trae SkiaSharp y sus assets nativos para Windows y Linux.
- Datos en vivo desde sgcWebSockets: añada **esegece.sgcGauges.sgcWebSockets** (edición Registered), que depende de esegece.sgcGauges y esegece.sgcWebSockets, véase Puente sgcWebSockets en .NET.

En C# todo host necesita `using esegece.sgcGauges;`, y `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUITextStyle** y los demás tipos de sgcUICore.

Véase también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos de .NET

Inicio rápido WinForms

Este inicio rápido construye en código un formulario con tres medidores WinForms: un velocímetro con una banda roja y un umbral a 200 km/h, una barra de combustible y una lámpara de exceso de velocidad que el umbral enciende y apaga.

Pasos

1. Cree un proyecto Windows Forms para `net8.0-windows` y añada el paquete **esegece.sgcGauges.WinForms** (**esegece.sgcGauges.WinForms.Community** en la edición Community).
2. Añada `using esegece.sgcGauges;` y `using esegece.sgcUICore;`.
3. Cree los controles y añádalos al formulario. Son controles WinForms normales: Bounds, Dock, Anchor y Enabled funcionan como siempre.
4. Defina primero el rango de la escala, después los rangos, los umbrales y el valor: un valor se limita a la escala cuando se escribe.
5. Asigne los eventos. Tienen los nombres de Delphi y se asignan con `=`.

Ejemplo

Notas

- **Value** es **Pointers[0].Value**. Cualquier hilo puede escribirlo: el control se repinta en el hilo de la interfaz y dispara **OnThreshold** allí.
- Con **Animation.Enabled** (el valor por defecto) la aguja se mueve al nuevo valor durante **Animation.Duration** milisegundos; el valor, los umbrales y los eventos cambian de inmediato. Solo se repinta el puntero en movimiento, como máximo **MaxFPS** fotogramas por segundo.
- **Theme** es por control; una propiedad de color dejada en 0 toma el color del tema.
- **SaveToPNG** y **SaveToSVG** escriben lo que muestra el control a su tamaño, con una escala opcional.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Enlace de datos en .NET
- Inicio rápido VCL

Inicio rápido WPF

Este inicio rápido declara en XAML un velocímetro, un display digital de rpm y una lámpara de exceso de velocidad, y enlaza los dos valores a un view model.

Pasos

1. Cree un proyecto WPF para `net8.0-windows` y añada el paquete **esegece.sgcGauges.Wpf** (**esegece.sgcGauges.Wpf.Community** en la edición Community).
2. Asigne el espacio de nombres en la ventana: `xmlns:sgc="clr-namespace:esegece.sgcGauges;assembly=esegece.sgcGauges.Wpf"`.
3. Declare los controles. Las propiedades simples como **Title**, **Units**, **ValueFormat** y **Digits** se definen como atributos; **Value** es una propiedad de dependencia y admite un enlace.
4. Rellene las colecciones (escalas, rangos, umbrales) y asigne los eventos en el code behind.

XAML

```
<Window x:Class="QuickStart.QuickStartWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:sgc="clr-namespace:esegece.sgcGauges;assembly=esegece.sgcGauges.Wpf"
        Title="sgcGauges" Width="680" Height="320">
  <StackPanel Orientation="Horizontal" Margin="8">
    <sgc:TsgcWpfRadialGauge x:Name="gauSpeed" Width="260" Height="260"
                            Title="Speed" Units="km/h" ValueFormat="0" Value="{Binding Speed}" />
    <StackPanel Margin="8,0,0,0">
      <sgc:TsgcWpfDigitalGauge x:Name="gauRpm" Width="340" Height="110"
                               Title="Engine" Units="rpm" ValueFormat="0" Digits="5" Value="{Binding Rpm}" />
      <sgc:TsgcWpfLEDGauge x:Name="ledAlarm" Width="160" Height="110" Title="Overspeed" Blink="True" />
    </StackPanel>
  </StackPanel>
</Window>
```

Code behind

Notas

- Cada control tiene una propiedad de dependencia: **ValueProperty** en los medidores radial, lineal, digital, LED, barra de LED, anillo de progreso y VU meter y **HeadingProperty** en la brújula. El reloj y el anunciador no tienen ninguna. Las demás propiedades son propiedades CLR. Véase Enlace de datos en .NET.
- **OnThreshold** se ejecuta en el hilo de la interfaz, también cuando el valor enlazado se define desde otro hilo.
- **Theme** es por control.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Enlace de datos en .NET
- Inicio rápido FMX

Inicio rápido Avalonia

Este inicio rápido construye en código una ventana Avalonia con un dial de media vuelta para una presión, un anillo de progreso y una barra de LED con bandas verde, ámbar y roja, todo en el tema oscuro. El mismo código funciona en Windows, Linux y macOS.

Pasos

1. Cree un proyecto `net8.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** (Avalonia 11.3) y añada **esegece.sgcGauges.Avalonia** (**esegece.sgcGauges.Avalonia.Community** en la edición Community).
2. Añada `using esegece.sgcGauges;` y `using esegece.sgcUICore;`.
3. Cree los controles, defina la escala y el valor y añádalos a un panel de la ventana.

Ejemplo

Notas

- Los controles también se pueden declarar en XAML con `xmlns:sgc="using:esegece.sgcGauges"`, por ejemplo `<sgc:TsgcAvaloniaRadialGauge Title="Pressure" Value="{Binding Pressure}" />`.
- **ValueProperty** (todos los controles), **ActiveProperty** (LED) y **HeadingProperty** (brújula) son propiedades con estilo que se enlazan en ambos sentidos por defecto, véase Enlace de datos en .NET.
- El temporizador de animación de un control se detiene cuando sale del árbol visual; los controles no necesitan Dispose.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Enlace de datos en .NET
- Inicio rápido FMX

Clases de .NET

Las clases de modelo tienen los nombres de Delphi y viven en el espacio de nombres **esegece.sgcGauges**. Los controles sustituyen el prefijo VCL y FMX por el nombre del host.

Medidores y controles

Modelo	Controles Delphi	WinForms	WPF	Avalonia
TsgcRadialGauge	TsgcVCLRadialGauge, TsgcFMXRadialGauge	TsgcWinFormsRadialGauge	TsgcWpfRadialGauge	TsgcAvaloniaRadialGauge
TsgcLinearGauge	TsgcVCLLinearGauge, TsgcFMXLinearGauge	TsgcWinFormsLinearGauge	TsgcWpfLinearGauge	TsgcAvaloniaLinearGauge
TsgcDigitalGauge	TsgcVCLDigitalGauge, TsgcFMXDigitalGauge	TsgcWinFormsDigitalGauge	TsgcWpfDigitalGauge	TsgcAvaloniaDigitalGauge
TsgcLEDGauge	TsgcVCLLEDGauge, TsgcFMXLEDGauge	TsgcWinFormsLEDGauge	TsgcWpfLEDGauge	TsgcAvaloniaLEDGauge
TsgcLEDBarGauge	TsgcVCLLEDBarGauge, TsgcFMXLEDBarGauge	TsgcWinFormsLEDBarGauge	TsgcWpfLEDBarGauge	TsgcAvaloniaLEDBarGauge
TsgcProgressRingGauge	TsgcVCLProgressRingGauge, TsgcFMXProgressRingGauge	TsgcWinFormsProgressRingGauge	TsgcWpfProgressRingGauge	TsgcAvaloniaProgressRingGauge
TsgcVUMeterGauge (Registered)	TsgcVCLVUMeterGauge, TsgcFMXVUMeterGauge	TsgcWinFormsVUMeterGauge	TsgcWpfVUMeterGauge	TsgcAvaloniaVUMeterGauge
TsgcClockGauge (Registered)	TsgcVCLClockGauge, TsgcFMXClockGauge	TsgcWinFormsClockGauge	TsgcWpfClockGauge	TsgcAvaloniaClockGauge
TsgcCompassGauge (Registered)	TsgcVCLCompassGauge, TsgcFMXCompassGauge	TsgcWinFormsCompassGauge	TsgcWpfCompassGauge	TsgcAvaloniaCompassGauge
TsgcAnnunciatorGauge (Registered)	TsgcVCLAnnunciatorGauge, TsgcFMXAnnunciatorGauge	TsgcWinFormsAnnunciatorGauge	TsgcWpfAnnunciatorGauge	TsgcAvaloniaAnnunciatorGauge

Los cuatro tipos Registered y sus controles no están en los ensamblados Community. Véase Tipos de medidor para lo que dibuja cada tipo.

Clases base

Clase	Paquete	Descripción
TsgcCustomGauge	esegece.sgcGauges	La base del modelo: **Scales**, **Pointers**, **Thresholds**, **Animation**, **Theme**, **Title**, **Units**, **ValueFormat**, **FormatSettings**, **Value**, **SetValue**, **ValueToText**, **BeginUpdate** y **EndUpdate**, **Render**,

		RenderStatic, **RenderDynamic**, **SaveToSVG** y los eventos **OnChange**, **OnThreshold** y **OnFormatValue**.
TsgcWinFormsCustomGauge	esegece.sgcGauges.WinForms	La base de los controles WinForms, sobre **TsgcUIWinFormsControl**.
TsgcWpfCustomGauge	esegece.sgcGauges.Wpf	La base de los controles WPF, sobre **TsgcUIWpfControl**.
TsgcAvaloniaCustomGauge	esegece.sgcGauges.Avalonia	La base de los controles Avalonia, sobre **TsgcUIAvaloniaControl**, con **ValueProperty**, **LogicalWidth()** y **LogicalHeight()**.
IsgcGaugeHost	esegece.sgcGauges	La implementa todo control: **GetGauge()** devuelve el modelo. El puente maneja cualquier objeto que la implemente.
sgcGauges_Raster	esegece.sgcGauges.Rendering	Los métodos de extensión **SaveToPNG** y **RenderToRaster** de **TsgcCustomGauge**.

Partes del modelo

Clase	Descripción
TsgcGaugeScale, TsgcGaugeScales	Una escala: **Minimum**, **Maximum**, **Mapping**, **MajorInterval**, **MinorCount**, **TickPlacement**, **LabelFormat**, **Ranges**.
TsgcGaugeRange, TsgcGaugeRanges	Una banda de color de una escala: **StartValue**, **EndValue**, **Color**, **Thickness**.
TsgcGaugePointer, TsgcGaugePointers	Un puntero de tipo aguja, marcador, barra o rango: **Kind**, **ScaleIndex**, **Value**, **EndValue**, **Origin**, **Color**, **NeedleStyle**, **MarkerShape**.
TsgcGaugeThreshold, TsgcGaugeThresholds	Un límite que dispara **OnThreshold**: **Value**, **PointerIndex**, **Direction**, **Hysteresis**, **Color**, **ValueColor**, **Enabled**, **Active**.
TsgcGaugeAnimation	La animación de los punteros: **Enabled**, **Duration** en milisegundos y **Easing**.
TsgcGaugeAnnunciatorTile, TsgcGaugeAnnunciatorTiles	Una casilla del anunciador: **Caption**, **State**, **Acknowledged**.
TsgcGaugeFormatSettings	Una struct con el **DecimalSeparator** y el **ThousandSeparator** de las etiquetas, '.' y ',' por defecto.

Delegados

Delegado	Firma
TsgcGaugeChangeEvent	`(object Sender, TsgcGaugeChangeKind aKind)`
TsgcGaugeThresholdEvent	`(object Sender, TsgcGaugeThreshold aThreshold, double aValue, bool aRising)`
TsgcGaugeFormatValueEvent	`(object Sender, int aPointer, double aValue, ref string aText)`
TsgcGaugeTileClickEvent	`(object Sender, int aIndex)` , el **OnTileClick** de los controles anunciador

Las clases del puente se enumeran en Puente sgcWebSockets en .NET.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- TsgcCustomGauge
- Tipos de medidor

Diferencias de .NET respecto a Delphi

La versión mantiene los nombres, los valores por defecto y el comportamiento de Delphi, así que los temas de Delphi de esta ayuda también valen para .NET. Estas son las diferencias que vienen del lenguaje y de los frameworks.

Lenguaje

- Los **eventos** son propiedades delegate con los nombres de Delphi (**OnThreshold**, **OnFormatValue**, **OnChange**, **OnTileClick**). Asígnelos con `=`: un manejador por evento, como en Delphi. El parámetro `var` de **OnFormatValue** es un parámetro `ref`. En los eventos de un control, Sender es el control.
- Las **funciones sin parámetros** siguen siendo métodos: **ActiveScaleCount()**, **ActivePointerCount()**, **NowMs()**, **GetGauge()**, **TsgcGaugeAnnunciatorTile.Flashing()**.
- **FormatSettings** es la struct **TsgcGaugeFormatSettings**: léala, cámbiela y asígnela de nuevo.
- Los **colores** son valores `uint` en orden ARGB (`0xFFE34948`). 0 significa el color del tema, como en Delphi.
- Las **colecciones** derivan de **TsgcGaugeCollection<T>**: un indexador (`Scales[0]`), **Add**, **Delete**, **Clear**, **Count**, **BeginUpdate** y **EndUpdate**, y `foreach`. **Assign(object)** copia una colección o un elemento.
- Las funciones libres de las units de Delphi son métodos estáticos: **sgcGauges_Types** (matemáticas y easing), **sgcGauges_Render** (geometría) y **sgcGauges_Raster** (PNG).
- No existe **Free**: los modelos son clases normales. Los controles WinForms y WPF siguen su framework; un control Avalonia detiene su temporizador al salir del árbol visual.

Ejemplo

Salida

- **SaveToSVG** es un método del modelo en esegece.sgcGauges. **SaveToPNG** de un modelo es un método de extensión en esegece.sgcGauges.Rendering, porque el PNG necesita el raster SkiaSharp. Los controles tienen ambos, a su propio tamaño.
- El SVG de un medidor tiene los mismos bytes que el SVG de Delphi del mismo medidor. El PNG lo dibuja SkiaSharp y puede diferir del PNG de Delphi en el antialiasing de los bordes.

Hilos y tiempo

- **SetValue** y **Value** son seguros para hilos, como en Delphi. Los controles se repintan y disparan **OnThreshold** en el hilo de la interfaz mediante el **SynchronizationContext** capturado al crearlos, en lugar de **TThread.Queue**. En un modelo sin control, **OnThreshold** se ejecuta en el hilo que escribió el valor.
- El reloj de la animación es un **Stopwatch** en lugar de **sgcGaugeNowMs**; **Clock** puede sustituirlo por cualquier **TsgcGaugeClockFunc**.

No portado

- Los auxiliares de streaming de .dfm (las funciones **Is*Stored**, las propiedades definidas ZeroValues y **GetNamePath**) no tienen equivalente en .NET. Las colecciones se rellenan en código.

Véase también

- Descripción general de .NET
- Clases de .NET
- Renderizado sin ventana en .NET
- TsgcCustomGauge
- Ediciones

Enlace de datos en .NET

Un medidor muestra un valor, así que el enlace de datos consiste en llevar un valor de un objeto de la aplicación al control. Cada host lo hace con su propio sistema de enlace, sobre la propiedad de valor del control. Las escalas, los rangos, los punteros y los umbrales son colecciones y se definen en código.

Host	Propiedades enlazables	Dirección por defecto
WinForms	**Value** (y toda propiedad pública) mediante **Control.DataBindings**	De la fuente al control con **DataSourceUpdateMode.Never**; los controles no disparan ningún evento de cambio.
WPF	Propiedades de dependencia **ValueProperty** (radial, lineal, digital, LED, barra de LED, anillo de progreso, VU meter) y **HeadingProperty** (brújula)	En un sentido. Con **BindingMode.TwoWay**, un valor que el modelo limita o ajusta circularmente se escribe de vuelta en la fuente.
Avalonia	Propiedades con estilo **ValueProperty** (todos los controles), **ActiveProperty** (LED) y **HeadingProperty** (brújula)	En ambos sentidos: un valor que el modelo limita o ajusta circularmente se escribe de vuelta en la fuente.

La fuente puede cambiar desde cualquier hilo: el modelo es seguro para hilos y el control se repinta en el hilo de la interfaz. Los propios enlaces de WinForms deben dispararse en el hilo de la interfaz, como en cualquier control WinForms.

WinForms

WPF

En XAML: `<sgc:TsgcWpfRadialGauge Value="{Binding Speed}" />`, véase Inicio rápido WPF. En código:

Avalonia

Feeds

Para manejar medidores con mensajes de WebSocket, MQTT o eventos enviados por el servidor sin un view model, enlácelos a una fuente de datos del puente, véase Puente sgcWebSockets en .NET.

Véase también

- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Puente sgcWebSockets en .NET

Renderizado sin ventana en .NET

Todo modelo de medidor se dibuja a sí mismo sin un control, así que un programa de consola, un servicio o un servidor web puede escribir imágenes PNG y SVG de medidores, en Windows, Linux y macOS. Esta es la forma .NET de Salida sin ventana.

Métodos

Método	Paquete	Descripción
SaveToSVG(Stream or string, int aWidth, int aHeight, double aScale = 1)	esegece.sgcGauges	Escribe un documento SVG del medidor. Sin dependencias nativas.
SaveToPNG(Stream or string, int aWidth, int aHeight, double aScale = 1)	esegece.sgcGauges.Rendering	Método de extensión de **sgcGauges_Raster**: dibuja en un raster SkiaSharp limpio de Round(aWidth * aScale) por Round(aHeight * aScale) píxeles y escribe un PNG.
RenderToRaster(int aWidth, int aHeight, double aScale = 1, double aNowMs = -1)	esegece.sgcGauges.Rendering	Devuelve el **TsgcUICanvas_Raster** con el medidor dibujado, para dibujar más o leer píxeles. El llamador lo libera.
Render(TsgcUICanvas aCanvas, int aWidth, int aHeight, double aScale = 1, double aNowMs = -1)	esegece.sgcGauges	Dibuja en cualquier canvas de sgcUICore.

Ejemplo

Notas

- Un renderizado sin ventana dibuja los valores de destino, nunca un fotograma de la animación, así que la salida del mismo medidor son siempre los mismos bytes. Un reloj con **Live** en false muestra su **Time**.
- aScale multiplica los píxeles y conserva la disposición: 200 por 200 a escala 2 es un PNG de 400 por 400 del mismo dibujo.
- En Linux, esegece.sgcGauges.Rendering trae los assets nativos de SkiaSharp; no hace falta ningún servidor X.

La demo ConsoleRender escribe un archivo PNG y otro SVG de cada tipo de medidor, véase Demos de .NET.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- Demos de .NET
- Salida sin ventana
- Inicio rápido sin ventana

Puente sgcWebSockets en .NET

El paquete **esegece.sgcGauges.sgcWebSockets** (edición Registered) es la versión .NET del puente sgcGaugesWS. Una fuente de datos se engancha a un cliente de sgcWebSockets .NET, lee cada mensaje como JSON y escribe los valores de sus enlaces en los medidores. El comportamiento es el descrito en Descripción general del puente.

Clase	Cliente	Lee
TsgcGaugeFeedSource_WebSocket	**Client**: un **TsgcWebSocketClient**, su evento **OnMessage**	Cada mensaje de texto. **ProcessMessage(string)** aplica un mensaje a mano.
TsgcGaugeFeedSource_MQTT	**Client**: un **TsgcWSProtocol_MQTT_Client**, su evento de publicación	El contenido de las publicaciones cuyo tema coincide con el filtro **Topic** de un enlace (comodines `+` y `#`). **ProcessPublish(topic, text)** aplica una a mano.
TsgcGaugeFeedSource_SSE	**Client**: un **TsgcHTTP1Client**, su evento **OnSSEMessage**	Los datos de los eventos cuyo tipo es igual al **EventName** de un enlace. **ProcessChunk(string)** y **ResetStream()** aplican un flujo a mano.

Toda fuente tiene **Active**, **Bindings**, **OnValue** y **OnError**. Un **TsgcGaugeFeedBinding** tiene **Gauge** (cualquier control, o cualquier objeto que implemente **IsgcGaugeHost**), **GaugeModel** (un modelo sin control), **PointerIndex**, **ValuePath**, **FilterPath** y **FilterValue**, **Topic**, **EventName** y **Active**.

Ejemplo

Notas

- Las fuentes nunca se conectan ni se suscriben: la aplicación conecta el cliente, suscribe los temas MQTT y llama a **GetSSE** del cliente HTTP.
- Los eventos del cliente son eventos .NET: una fuente añade su propio manejador con `+=` y quita solo ese, de modo que los manejadores de la aplicación siguen funcionando y se ejecutan en el orden en que se añadieron.
- Los valores se escriben en el hilo del cliente. Los medidores son seguros para hilos y se repintan en el hilo de la interfaz; **OnValue** y **OnError** se ejecutan en el hilo del cliente, así que publique en el hilo de la interfaz antes de tocar un control.
- Un enlace vigila el evento **Disposed** de un control WinForms y borra **Gauge** cuando el control desaparece. Para los controles WPF y Avalonia y para **GaugeModel**, borre el enlace antes de que desaparezca el medidor.
- Una cadena numérica se lee con '.' como separador decimal, sea cual sea la cultura; true y false se leen como 1 y 0.

La demo MQTTLive ejecuta seis medidores sobre MQTT con un pequeño broker dentro del proceso, véase Demos de .NET.

Véase también

- Descripción general del puente
- Fuente MQTT
- Fuente WebSocket

- Fuente de eventos enviados por el servidor
- Enlace de datos en .NET
- Demos de .NET

Demos de .NET

Las demos de .NET están en `demos\Gauges` de la carpeta .NET de sgcUI, una carpeta por demo con un archivo de proyecto, Program.cs y un README.md. La interfaz se construye en código.

Demo	Host	Muestra
Gallery	WinForms	Los diez controles de medidor en un formulario: cambio de tema, animación activada y desactivada, una simulación de valor y exportación PNG al doble de tamaño. El umbral de velocidad a 200 km/h activa un LED de alarma y la casilla HIGH SPEED; un clic en una casilla la reconoce.
GalleryWpf	WPF	La misma galería con los controles TsgcWpf.
GalleryAvalonia	Avalonia	La misma galería con los controles TsgcAvalonia.
HMIPanel	WinForms	Un proceso simulado con dos depósitos y dos bombas: medidores lineales de nivel, medidores radiales de presión, displays digitales de caudal, LED de las bombas y un anunciador activado por un único manejador **OnThreshold** compartido.
MQTTLive	WinForms	Seis medidores alimentados por **TsgcGaugeFeedSource_MQTT**. Con la opción de loopback, Connect inicia un pequeño broker MQTT en el proceso, así que la demo no necesita broker ni red. Solo en la edición Registered.
ConsoleRender	Consola	Un modelo de cada tipo escrito en archivos PNG y SVG, sin framework de interfaz; el equivalente de la demo Headless de Delphi.

Compilar y ejecutar

```
dotnet run --project Gallery\Gallery.csproj
```

Dentro del árbol de código de sgcUI, una demo se compila contra los proyectos de la biblioteca en la edición Registered; añada `-p:SgcCommunity=true` para la edición Community. En el zip de la demo, esta restaura los paquetes desde nuget.org, por defecto los paquetes Community; defina `-p:SgcGaugesEdition=Registered` para los Registered. En la edición Community, las galerías muestran seis medidores, HMIPanel lista las alarmas en un cuadro de lista en lugar del anunciador y ConsoleRender escribe los seis tipos Community. MQTTLive necesita la edición Registered y el paquete esegece.sgcWebSockets.

Autoprueba

Toda demo acepta `--test result.txt`: ejecuta una autocomprobación fuera de pantalla, escribe una línea PASS o FAIL por comprobación y termina con el número de comprobaciones fallidas. `--snapshot file.png` guarda una imagen PNG y termina.

Véase también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Renderizado sin ventana en .NET
- Puente sgcWebSockets en .NET

Solución de problemas

El valor se detiene en 100

Un valor se limita al rango de su escala cuando se escribe, y la escala por defecto va de 0 a 100. Defina Minimum y Maximum de la escala antes del valor. Un cambio posterior del rango vuelve a limitar los valores almacenados, pero no restaura un valor que fue recortado.

La segunda escala o puntero no se dibuja

La edición Community dibuja solo la primera escala y el primer puntero. Los demás se conservan en las colecciones. Véase Ediciones.

TsgcVCLClockGauge o TsgcCompassGauge no está declarado

El VU meter, el reloj, la brújula y el anunciador forman parte de la edición Registered. En Community sus units están vacías y sus controles no se registran.

Una pantalla de 14 segmentos muestra 7 segmentos

Las pantallas de 14 segmentos y de matriz de puntos forman parte de la edición Registered. En Community se dibujan como 7 segmentos.

La aguja salta en lugar de moverse

Compruebe Animation.Enabled y Animation.Duration. No hay animación en tiempo de diseño, para valores cargados del archivo de formulario, para `SetValue(aPointer, aValue, False)` y en un renderizado sin ventana, que dibuja el valor de destino.

OnThreshold no se dispara

- Con gtdRising y gtdFalling, el evento se dispara solo cuando el umbral pasa a estar activo. Use gtdBoth para recibir también el evento de fin.
- Tras un evento, el valor debe salir de la banda dada por Hysteresis antes de que el umbral pueda dispararse de nuevo.
- Un cambio del propio umbral actualiza Active sin evento.
- Compruebe Enabled y PointerIndex. En Community, los umbrales de los punteros posteriores al primero no se evalúan.

No hay marca de umbral en la escala

Un umbral solo dibuja una marca cuando su Color no es 0.

Las etiquetas muestran '.' en una configuración regional con coma decimal

Los medidores usan su propio FormatSettings, con '.' por defecto, de modo que la salida no depende de la máquina. Asigne FormatSettings del modelo, mediante la propiedad Gauge de un control, para cambiarlo. Véase Temas.

El texto es demasiado pequeño o demasiado grande

Los tamaños de fuente son relativos al tamaño del medidor, de modo que el texto sigue el tamaño del control. Cambie el Size de TitleFont, ValueFont o LabelFont para cambiar la proporción.

Escritura de valores desde un hilo de trabajo

Value, SetValue y el Value y el EndValue del puntero son seguros para hilos, y los controles se repintan y llaman a OnThreshold en el hilo principal. Todas las demás propiedades pertenecen al hilo principal.

Ejemplo

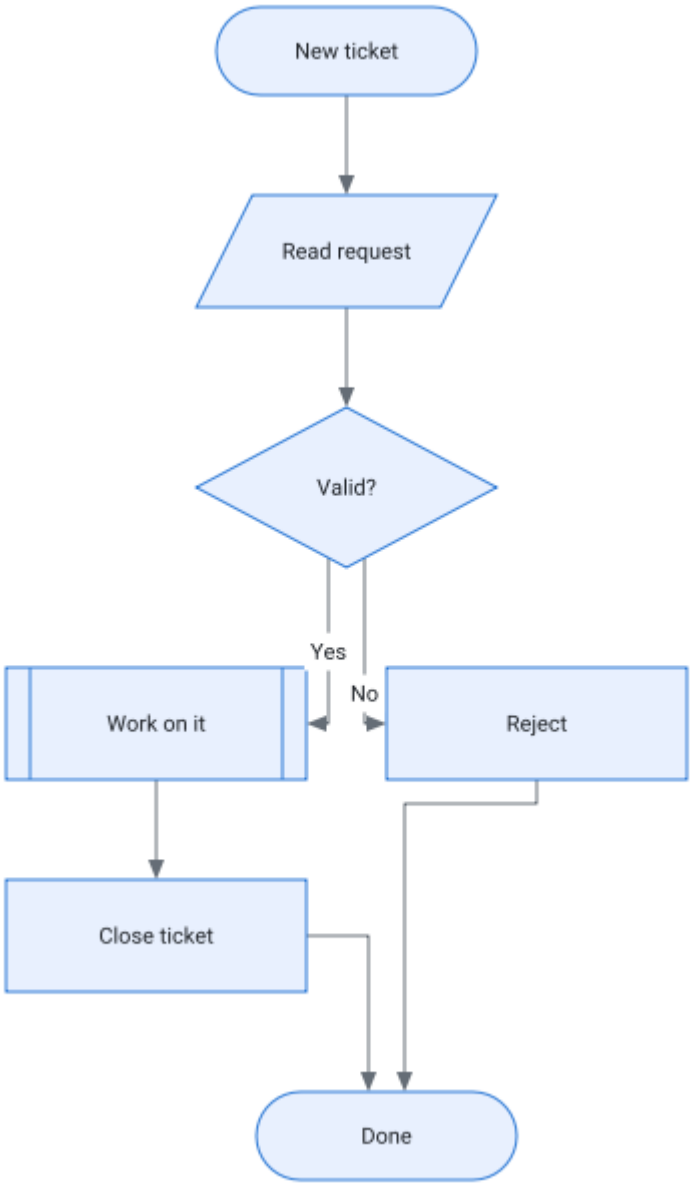
```
uses
  SysUtils, Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

type
  TSensorThread = class(TThread)
  private
    FGauge: TsgcVCLRadialGauge;
  protected
    procedure Execute; override;
  public
    constructor Create(aGauge: TsgcVCLRadialGauge);
  end;

constructor TSensorThread.Create(aGauge: TsgcVCLRadialGauge);
begin
  FGauge := aGauge;
  inherited Create(False);
end;

procedure TSensorThread.Execute;
begin
  while not Terminated do
  begin
    // Value is thread safe: no Synchronize needed
    FGauge.Value := Random(100);
    Sleep(250);
  end;
end;
```

Véase también

- Ediciones
- Animación
- Umbrales

Introducción

sgcDiagram añade un editor y visor de diagramas a las aplicaciones Delphi y C++Builder, para VCL y FireMonkey: diagramas de flujo, organigramas y grafos de nodos. El usuario arrastra formas desde una paleta, dibuja conectores entre ellas, mueve, redimensiona y gira los nodos, edita su texto en el propio lugar y deshace cada paso. El mismo diagrama se puede construir, organizar y exportar por código, sin ventana, en un servidor o en Linux64.
Todo vive en un único objeto de documento, **TsgcDiagramDocument**: los nodos, los conectores con sus puertos y etiquetas, las capas, las imágenes y los estilos predeterminados. **TsgcVCLDiagramView** y **TsgcFMXDiagramView** muestran y editan un documento, **TsgcVCLDiagramPalette** y **TsgcFMXDiagramPalette** ofrecen las formas para arrastrar a la vista, y **TsgcDiagramRenderer** escribe un documento en PNG y SVG sin una vista. Un documento se guarda y se carga como JSON, y una vista almacena su diagrama en el archivo del formulario.
sgcDiagram forma parte de la suite visual de eSeGeCe. Se dibuja con **sgcUICore**, el paquete de runtime compartido con el canvas, el rasterizador, las fuentes, el escritor de SVG y los temas claro y oscuro, de modo que un diagrama tiene el mismo aspecto en VCL, en FireMonkey y en los archivos exportados. sgcDiagram requiere sgcUICore, que se instala junto con él.



Ediciones

sgcDiagram se distribuye como Registered, Trial y Community. La edición Community es un editor manual completo: todas las formas, rutas personalizadas, imágenes, los tres tipos de conector, puertos y anclaje, grupos y contenedores, deshacer, el portapapeles, JSON, PNG y SVG, temas, ajuste y guías, la paleta y el visor de solo lectura. Deja fuera los diseños automáticos, el enrutamiento ortogonal con esquiva de obstáculos, el enlace de datos y más de una capa. Consulte Ediciones.

Por dónde empezar

- Qué es sgcDiagram: los tipos de diagrama que dibuja y lo que el usuario puede hacer con ellos.
- Arquitectura: el documento, las vistas, el controlador y las unidades.
- Inicio rápido VCL, Inicio rápido FMX, Inicio rápido C++Builder y Inicio rápido sin ventana: un primer diagrama en pocos minutos.
- Los componentes: TsgcVCLDiagramView, TsgcFMXDiagramView, TsgcVCLDiagramPalette, TsgcFMXDiagramPalette, TsgcDiagramDataBinding y sus Eventos.
- Las clases: TsgcDiagramDocument, TsgcDiagramNode, TsgcDiagramConnector, TsgcDiagramPort, TsgcDiagramStyle, TsgcDiagramLayer, TsgcDiagramController, TsgcDiagramRenderer y el Formato JSON.
- Diseño por capas, Diseño en árbol y Enlace de datos: organizan un diagrama automáticamente, o lo construyen a partir de un dataset.
- Las características: Formas, Conectores y enrutamiento, Puertos y anclaje, Edición y teclado, Ajuste y guías, Deshacer y portapapeles, Capas, Serialización, Exportación, Temas y Modo visor.
- Plataformas: los IDE, los frameworks y los sistemas operativos compatibles.
- Solución de problemas: los problemas comunes y sus causas.

Qué es sgcDiagram

sgcDiagram es un conjunto de componentes de diagrama: un documento de nodos y conectores, una vista interactiva que lo edita, una paleta de formas y un exportador. Está pensado para los diagramas que muestran las aplicaciones de negocio y que dejan editar a sus usuarios.

Qué se puede dibujar

- **Diagramas de flujo.** Formas de proceso, decisión, terminador, datos, documento, proceso predefinido y círculo conector, conectores con flechas y etiquetas Sí y No.
- **Organigramas y árboles.** Cuadros con un nombre y un cargo, unidos por conectores ortogonales. La edición Registered los organiza con un diseño en árbol y los construye a partir de un dataset.
- **Grafos de nodos.** Nodos con puertos tipados de entrada y salida, conectores curvos y su propio dibujo en OnDrawNode, como en un editor de nodos o en un diseñador de canalizaciones.
- **Bocetos libres.** Rectángulos, rectángulos redondeados, elipses, texto libre, imágenes, formas personalizadas a partir de datos de ruta SVG y contenedores que contienen otros nodos.

Qué puede hacer el usuario

- Arrastrar una forma desde la paleta a la vista, o elegirla y dibujarla con la herramienta de forma.
- Seleccionar con un clic, clics con Shift y Ctrl o un rectángulo de selección, y luego mover, redimensionar, girar, ajustar con las flechas, agrupar, eliminar, copiar y pegar.
- Dibujar un conector desde un puerto, anclar sus extremos a puertos o al contorno de un nodo, mover sus etiquetas y sus waypoints.
- Editar el texto de un nodo en el lugar: doble clic, F2 o simplemente empezar a escribir.
- Deshacer y rehacer cada acción, desplazarse con la rueda del ratón, el botón central o la barra espaciadora, hacer zoom con Ctrl y la rueda o un gesto de pellizco.

Qué puede hacer el código

- Construir un diagrama con AddNode y AddConnector, cambiar cualquier propiedad y agrupar los cambios en un paso de deshacer con BeginTransaction y EndTransaction.
- Guardarlo y cargarlo como JSON, como archivo, stream o cadena.
- Exportarlo a PNG y SVG con TsgcDiagramRenderer, también en una aplicación de consola o en Linux64.
- Escuchar OnChange, OnSelectionChange, OnItemClick, OnConnecting y los demás eventos para reaccionar al usuario.

Lo que no hace

La versión 1 no tiene plantillas BPMN, UML ni de red, ni pools de carriles, ni diseños dirigidos por fuerzas o circulares, ni exportación a PDF, ni intercambio de portapapeles con otras aplicaciones, ni minimapa, reglas, barras de desplazamiento o impresión, ni importación de SVG o Visio, ni texto enriquecido en los nodos.

Consulte también

- Arquitectura
- Ediciones
- Inicio rápido VCL

Arquitectura

sgcDiagram separa el documento, la interacción y el dibujo, de modo que el mismo código edita un diagrama en VCL, en FireMonkey y en una prueba sin ventana.

Parte	Función
TsgcDiagramDocument	El modelo: Nodes, Connectors, Layers, Images, Selection, los NodeStyle y ConnectorStyle predeterminados, Routing, GridSize, Theme y la pila de deshacer. Cada cambio se notifica una vez por cada par BeginUpdate y EndUpdate mediante OnChange.
TsgcDiagramNode	Un cuadro con Shape, Left, Top, Width, Height, Angle, Text, Style, Ports, Layer, ContainerId, GroupId, ImageKey, PathData, Locked, Visible, Data y Tag. El orden z es el índice en Nodes.
TsgcDiagramConnector	Un enlace entre dos nodos o dos puntos libres: Kind, SourceNodeId, SourcePortId, TargetNodeId, TargetPortId, SourcePoint, TargetPoint, Waypoints, StartArrow, EndArrow, ArrowSize, Labels y Style. Points contiene la ruta calculada.
Router	Calcula la ruta de cada conector cuyos extremos, nodos o tipo cambiaron, al final de cada conjunto de cambios cuando Routing.AutoRoute es True.
TsgcDiagramController	La máquina de estados de la interacción: prueba de impacto, selección, arrastre, redimensionamiento, rotación, conexión, ajuste, guías, edición de texto, zoom y desplazamiento. Es independiente del framework y trabaja en píxeles del dispositivo por fuera y en unidades del mundo por dentro.
TsgcDiagramPainter y la caché de la vista	Dibujan el documento en un TsgcUICanvas de sgcUICore. La vista mantiene una imagen en caché del contenido, de modo que un desplazamiento es una copia de píxeles y una edición repinta solo su rectángulo sucio.
TsgcVCLDiagramView, TsgcFMXDiagramView	Los controles. Poseen un documento (la propiedad Diagram) y un Controller, y traducen el ratón, el teclado, los gestos y el arrastrar y soltar en llamadas al controlador.
TsgcVCLDiagramPalette, TsgcFMXDiagramPalette	Una cuadrícula de elementos de forma agrupados por categoría. Un clic elige la herramienta de forma de la vista enlazada, un arrastre suelta la forma sobre ella.
TsgcDiagramRenderer	Exporta un documento a PNG y SVG sin una vista.

Coordenadas

Las unidades del mundo son píxeles con zoom 1 y 96 PPI, con y creciendo hacia abajo. Los ángulos de los nodos están en grados, en sentido horario, alrededor del centro del nodo. Un color es un TsgcUIColor, y 0 significa el color del tema. ScreenToDiagram y DiagramToScreen de la vista convierten entre los píxeles del dispositivo del control y las unidades del mundo.

Notificaciones de cambio

El documento recoge los cambios entre el BeginUpdate y el EndUpdate más externos y los notifica una vez: primero a la vista que lo posee, que repinta el rectángulo sucio, y luego mediante OnChange con un conjunto de valores TsgcDiagramChange (dgkGeometry, dgkStyle, dgkStructure, dgkSelection, dgkView) y OnSelectionChange. Un cambio hecho por código fuera de BeginUpdate y EndUpdate se notifica de inmediato.

Unidades

Unidad	Contenido
sgcDiagram_Types	Las enumeraciones, TsgcDiagramPoint, TsgcDiagramRect, los auxiliares de geometría como sgcDiagramPoint y sgcDiagramBounds, y EsgcDiagramError.
sgcDiagram_Shapes	El contorno, el rectángulo de texto y los puertos predeterminados de cada forma, y el analizador de datos de ruta SVG.
sgcDiagram_Model	TsgcDiagramDocument, los nodos, conectores, puertos, etiquetas, capas, imágenes, estilos y la selección.
sgcDiagram_Undo, sgcDiagram_JSON, sgcDiagram_Router	La pila de deshacer, el formato JSON y el portapapeles, el enrutamiento de conectores.
sgcDiagram_Render	El painter, la caché de la vista y TsgcDiagramRenderer.
sgcDiagram_Palette, sgcDiagram_Controller	El modelo de la paleta y el controlador de interacción.
sgcDiagram_Layout, sgcDiagram_Data	Los diseños por capas y en árbol y el enlace de datos. Edición Registered; unidades vacías en Community.
sgcDiagram_VCL, sgcDiagram_FMX	La vista y la paleta VCL y FireMonkey.

Consulte también

- Qué es sgcDiagram
- Plataformas
- Deshacer y portapapeles
- TsgcDiagramDocument
- TsgcDiagramController
- TsgcDiagramRenderer

Ediciones

sgcDiagram se compila a partir del mismo código fuente para todas las ediciones.

Edición	Incluye
Registered	Todo: las vistas y paletas para VCL y FireMonkey, todas las formas, los tres tipos de conector, puertos, grupos y contenedores, deshacer, el portapapeles, JSON, PNG y SVG, temas, ajuste, guías y las características avanzadas que se indican abajo. La página de paleta SGC Diagram muestra 5 componentes.
Trial	Las mismas características que Registered. Una compilación con SGC_TRIAL muestra un mensaje de que es una versión de prueba cuando se crea un componente después de que haya pasado la ventana de lanzamiento indicada en la compilación. Nada más cambia.
Community	Todo excepto las características avanzadas: un editor manual completo de diagramas de flujo y organigramas para VCL, FireMonkey y uso sin ventana. La página de paleta SGC Diagram muestra 4 componentes (las dos vistas y las dos paletas).

Las características avanzadas

Las características avanzadas se compilan bajo el define SGC_DIAGRAM_ADVANCED, que sgcVer.inc desactiva en la edición Community. Las listas de unidades de los paquetes son las mismas en todas las ediciones: sgcDiagram_Layout y sgcDiagram_Data se compilan como unidades vacías en Community.

Característica	En la edición Community
Diseño automático: TsgcDiagramLayeredLayout y TsgcDiagramTreeLayout	No disponible. Los nodos permanecen donde el usuario o el código los coloca.
Enrutamiento ortogonal con esquiva de obstáculos y separación de segmentos paralelos	Los conectores dgcOrthogonal se enrutan como codos simples que no esquivan los nodos, y RouteFailed permanece en False.
Enlace de datos: TsgcDiagramDataBinding construye un diagrama a partir de un dataset	No disponible.
Capas además de la primera	Layers.Add genera EsgcDiagramError cuando ya existe una capa. Un archivo guardado por la edición Registered conserva los nombres de sus capas y se carga y se guarda sin cambios, pero cada capa se dibuja, se exporta y se edita como visible y desbloqueada.

Un archivo de diagrama es el mismo en todas las ediciones, de modo que un documento creado con la edición Registered se abre en una compilación Community y al revés.

Consulte también

- Conectores y enrutamiento
- Capas
- Plataformas
- Diseño por capas
- Diseño en árbol
- TsgcDiagramDataBinding

Plataformas

sgcDiagram es un único árbol de código fuente para todos los IDE y plataformas. El documento, el enrutador, el controlador, el painter y el renderizador usan solo la RTL y sgcUICore, de modo que son el mismo código en todos los destinos; solo las vistas y las paletas están ligadas a un framework.

Destino	Compatibilidad
Delphi VCL	Delphi 7 a Delphi 13, Win32 y Win64: TsgcVCLDiagramView y TsgcVCLDiagramPalette.
Delphi FireMonkey	Delphi XE6 a Delphi 13 para Windows, macOS, iOS y Android: TsgcFMXDiagramView y TsgcFMXDiagramPalette. Se compila bajo SGC_DIAGRAM_FMX.
Linux64	Delphi 10.2 y posteriores, compilado bajo SGC_DIAGRAM_LINUX: el documento, JSON, enrutamiento, diseños y TsgcDiagramRenderer, de modo que un servidor puede construir un diagrama y producir archivos PNG y SVG. Consulte Inicio rápido sin ventana.
C++Builder	Los mismos componentes con los encabezados .hpp generados. Consulte Inicio rápido C++Builder.
.NET	Un port en C# con la misma API para WinForms, WPF y Avalonia, la vista y la paleta en cada host y la salida headless de JSON, PNG y SVG, en .NET Framework 4.7.2 y .NET 8 y 9, consulte Descripción general de .NET.

Unidades por framework

Unidades	Framework
sgcDiagram_Const, sgcDiagram_Types, sgcDiagram_Shapes, sgcDiagram_Model, sgcDiagram_Undo, sgcDiagram_JSON, sgcDiagram_Router, sgcDiagram_Layout, sgcDiagram_Render, sgcDiagram_Palette, sgcDiagram_Controller	Ninguno. El modelo, los motores, el painter y el formato de archivo.
sgcDiagram_Data	Solo la unidad DB (dbrtl), ninguna unidad VCL ni FireMonkey.
sgcDiagram_VCL	VCL, solo Windows.
sgcDiagram_FMX	FireMonkey, Delphi XE6 y posteriores.

Paquetes

sgcDiagram incluye un paquete de runtime independiente del framework y un paquete de runtime por framework, de modo que una aplicación FireMonkey para Android, iOS o macOS nunca enlaza la VCL. Cada nombre de paquete termina con el sufijo del IDE, por ejemplo sgcDiagramD13.dpk para Delphi 13 y sgcDiagramC13.cbproj para C++Builder 13.

Paquete	Tipo	Contenido	Requiere
sgcDiagram	Runtime, indepen-	Todas las unidades del producto que no usan ni la VCL ni FireMonkey: el documento, las formas, deshacer, JSON, el enrutador, las disposiciones, el renderizador, el modelo de la	rtl, dbrtl, sgcUICore

	diente del framework	paleta, el controlador y, fuera de la edición Community, TsgcDiagramDataBinding (sgcDiagram_Data).	
sgcDiagramVCL	Runtime, VCL	sgcDiagram_VCL (TsgcVCLDiagramView, TsgcVCLDiagramPalette)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcDiagram
sgcDiagramFMX	Runtime, FireMonkey, desde XE6	sgcDiagram_FMX (TsgcFMXDiagramView, TsgcFMXDiagramPalette)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcDiagram
dclsgcDiagram	Tiempo de diseño, VCL	Registra TsgcVCLDiagramView, TsgcVCLDiagramPalette y, fuera de la edición Community, TsgcDiagramDataBinding en la página de paleta **SGC Diagram**, y añade los verbos "Load Diagram...", "Save Diagram..." y "Clear Diagram" al editor de componente de la vista.	designide, rtl, vcl, sgcDiagram, sgcDiagramVCL
dclsgcDiagramFMX	Tiempo de diseño, FireMonkey, desde XE6	Registra TsgcFMXDiagramView y TsgcFMXDiagramPalette en la página de paleta **SGC Diagram**, con los mismos verbos.	designide, rtl, fmx, sgcDiagram, sgcDiagramFMX, dclsgcDiagram

sgcUICore sigue la misma división: sgcUICore, sgcUICoreVCL y sgcUICoreFMX, sin paquete de tiempo de diseño. dclsgcDiagramFMX requiere dclsgcDiagram, así que instale primero dclsgcDiagram. sgcDiagram no tiene paquete puente con sgcWebSockets.

Versiones del IDE y plataformas de destino

Delphi	sgcDiagram	sgcDiagramVCL	sgcDiagramFMX	dclsgcDiagram, dclsgcDiagramFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Ninguna	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Ninguna	Win32
XE6 a 10.2	Win32, Win64	Win32, Win64	Win32, Win64	Win32
10.3	Win32, Win64, Linux64, OSX64	Win32, Win64	Win32, Win64, OSX64	Win32
10.4	Win32, Win64, Linux64, OSX64, Android, Android64, iOSDevice64	Win32, Win64	Win32, Win64, OSX64, Android, Android64, iOSDevice64	Win32
11	Win32, Win64, Linux64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32, Win64	Win32, Win64, OSX64, OSXARM64, Android, Android64, iOSDevice64, iOSSimARM64	Win32
12, 13	Igual que 11	Win32, Win64	Igual que 11	Win32, Win64

De XE6 a 10.2 el paquete FireMonkey se compila solo para Windows. El IDE carga los paquetes de tiempo de diseño, por lo que se compilan solo para la plataforma del IDE.

C++Builder	sgcDiagram, sgcDiagramVCL	sgcDiagramFMX	dclsgcDiagram, dclsgcDiagramFMX
2007, 2010, XE, XE2	Win32	Ninguna	Win32 (solo dclsgcDiagram)
XE3 a XE5	Win32, Win64	Ninguna	Win32 (solo dclsgcDiagram)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Los proyectos de C++Builder tienen los mismos nombres con el sufijo C, por ejemplo sgcDiagramC13.cbproj, sgcDiagramVCLC13.cbproj, sgcDiagramFMXC13.cbproj, dclsgcDiagramC13.cbproj y dclsgcDiagramFMXC13.cbproj. Los paquetes de C++Builder se compilan solo para Windows. Las notas de C++Builder que se aplican a todos los productos sgcUI (paquetes de runtime en Win64, C++Builder 10.3 en Win64, nombres de archivo de los paquetes Win64x) están en el tema Plataformas y paquetes de la ayuda de la suite sgcUI.

FireMonkey en Android, iOS y macOS

Los controles FireMonkey (TsgcFMXDiagramView y TsgcFMXDiagramPalette) tienen como destino macOS desde Delphi 10.3, Android e iOS desde Delphi 10.4, y macOS ARM y el simulador de iOS desde Delphi 11. En un proyecto FireMonkey para esas plataformas:

- Use los controles FireMonkey. Los controles VCL y sgcDiagramVCL solo funcionan en Windows.
- Coloque el control en un formulario, o añada su unidad a la cláusula uses: sgcDiagram_FMX.
- Seleccione la plataforma de destino en la ventana Proyectos. El instalador añade la carpeta Source de sgcUI y las unidades compiladas de cada plataforma (LibD<version>\<platform>) a la ruta de biblioteca de cada plataforma que compila. Sin el instalador, añada la carpeta Source de sgcUI a la ruta de biblioteca de la plataforma de destino (Herramientas, Opciones, Lenguaje, Delphi, Biblioteca).
- Las aplicaciones Android e iOS se enlazan de forma estática, por lo que no se distribuye ningún paquete de runtime.

Linux64

El paquete independiente del framework sgcDiagram se compila para Linux64 desde Delphi 10.3. Una aplicación Linux64, por ejemplo un programa de consola o un servidor, usa el documento, JSON, el enrutamiento, las disposiciones y TsgcDiagramRenderer para construir un diagrama y producir archivos PNG y SVG. No se compila ningún paquete VCL ni FireMonkey para Linux64, porque FireMonkey en Linux necesita FMX para Linux (FMXLinux), un complemento de terceros.

Notas por plataforma

- **DPI alto.** La vista VCL se adapta al PPI del monitor y la vista FireMonkey a la escala del formulario. El zoom 1 muestra una unidad del mundo por píxel independiente del dispositivo.
- **Táctil.** En FireMonkey, un gesto de pellizco hace zoom alrededor de su centro y un dedo en un espacio vacío desplaza la vista. Un dedo sobre un nodo lo arrastra, también en una pantalla táctil de Windows.
- **Arrastrar y soltar desde la paleta.** En iOS y Android no hay servicio de arrastre: toque un elemento de la paleta y luego toque la vista para colocar la forma.
- **Tecla Tab.** Tab y Shift+Tab recorren la selección por los elementos en VCL. Un formulario FireMonkey gestiona Tab antes que el control, de modo que las teclas mueven allí el foco.
- **Delphi 7 y Delphi 2007.** Estas versiones tienen cadenas Ansi, de modo que el texto fuera de la página de códigos del sistema se pierde cuando se carga un archivo JSON.
- **Ediciones.** SGC_DIAGRAM_ADVANCED, que sgcVer.inc desactiva en la edición Community, decide las características avanzadas en todas las plataformas. Consulte Ediciones.

Consulte también

- Arquitectura
- Exportación
- Solución de problemas

Inicio rápido VCL

Instale los paquetes sgcDiagram para su versión de Delphi. Los componentes aparecen en la página de paleta **SGC Diagram**.

En tiempo de diseño

- Suelte un **TsgcVCLDiagramPalette** en el formulario y establezca Align en alLeft.
- Suelte un **TsgcVCLDiagramView** y establezca Align en alClient.
- Establezca la propiedad View de la paleta en la vista.
- Ejecute la aplicación. Arrastre formas desde la paleta, dibuje conectores desde los puertos que aparecen cuando el puntero está sobre un nodo, haga doble clic en un nodo para editar su texto.

El menú contextual de la vista en tiempo de diseño tiene verbos para cargar un diagrama desde un archivo JSON, guardarlo y borrarlo. El diagrama se almacena en el dfm como la propiedad DiagramJSON.

En el código

```
uses
  Forms, Controls,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

type
  TFormMain = class(TForm)
  private
    FPalette: TsgcVCLDiagramPalette;
    FView: TsgcVCLDiagramView;
  public
    procedure CreateDiagram;
  end;

procedure TFormMain.CreateDiagram;
var
  oStart, oCheck, oDone: TsgcDiagramNode;
begin
  FPalette := TsgcVCLDiagramPalette.Create(Self);
  FPalette.Parent := Self;
  FPalette.Align := alLeft;
  FPalette.Width := 180;
  FView := TsgcVCLDiagramView.Create(Self);
  FView.Parent := Self;
  FView.Align := alClient;
  FPalette.View := FView;
  // three nodes and two connectors, one undo step
  FView.Diagram.BeginTransaction('Create');
  try
    oStart := FView.Diagram.AddNode(dgsTerminator, 40, 40, 140, 50, 'Start');
    oCheck := FView.Diagram.AddNode(dgsDecision, 30, 140, 160, 90, 'Valid?');
    oDone := FView.Diagram.AddNode(dgsProcess, 40, 290, 140, 60, 'Save');
    FView.Diagram.AddConnector(oStart, oCheck);
    FView.Diagram.AddConnector(oCheck, oDone).Labels.Add.Text := 'Yes';
  finally
    FView.Diagram.EndTransaction;
  end;
  FView.ZoomToFit;
end;
```

Próximos pasos

- Guarde el diagrama con FView.Diagram.SaveToFile, consulte Serialización.

- Expórtelo con FView.SaveToPNG y FView.SaveToSVG, consulte Exportación.
- Cambie la vista a un visor de solo lectura con ReadOnly, consulte Modo visor.

Consulte también

- Inicio rápido FMX
- Inicio rápido sin ventana
- Edición y teclado
- TsgcVCLDiagramView
- TsgcVCLDiagramPalette

Inicio rápido FMX

Los componentes FireMonkey son **TsgcFMXDiagramView** y **TsgcFMXDiagramPalette**, en la página de paleta **SGC Diagram** de Delphi XE6 y posteriores. Tienen las mismas propiedades, métodos y eventos que los VCL, y la misma clase de documento, de modo que el código que trabaja sobre la propiedad Diagram se comparte.

En tiempo de diseño

- Suelte un TsgcFMXDiagramPalette con Align Left y un TsgcFMXDiagramView con Align Client.
- Establezca la propiedad View de la paleta en la vista.
- Ejecute la aplicación en Windows o macOS y arrastre formas a la vista. En iOS y Android, toque un elemento de la paleta y luego toque la vista.

En el código

```
uses
  FMX.Forms, FMX.Types,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_FMX;

type
  TFormMain = class(TForm)
  private
    FPalette: TsgcFMXDiagramPalette;
    FView: TsgcFMXDiagramView;
  public
    procedure CreateDiagram;
  end;

procedure TFormMain.CreateDiagram;
var
  oA, oB: TsgcDiagramNode;
begin
  FPalette := TsgcFMXDiagramPalette.Create(Self);
  FPalette.Parent := Self;
  FPalette.Align := TAlignLayout.Left;
  FPalette.Width := 180;
  FView := TsgcFMXDiagramView.Create(Self);
  FView.Parent := Self;
  FView.Align := TAlignLayout.Client;
  FPalette.View := FView;
  oA := FView.Diagram.AddNode(dgsRoundRect, 40, 40, 140, 60, 'Order');
  oB := FView.Diagram.AddNode(dgsRoundRect, 260, 40, 140, 60, 'Invoice');
  FView.Diagram.AddConnector(oA, oB);
  FView.ZoomToFit;
end;
```

Táctil

- Un gesto de pellizco hace zoom alrededor de su centro.
- Un dedo en un espacio vacío desplaza la vista, un dedo sobre un nodo lo arrastra.

Consulte también

- Inicio rápido VCL
- Plataformas
- TsgcFMXDiagramView
- TsgcFMXDiagramPalette

Inicio rápido C++Builder

Instale los paquetes sgcDiagram para su versión de C++Builder, añada `sgcUI\Source` y la carpeta de las unidades compiladas, por ejemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), a las rutas de include y de biblioteca del proyecto, y use los componentes de la página de paleta **SGC Diagram** exactamente como en Delphi.

Qué tener en cuenta en C++

- Las enumeraciones no tienen ámbito: escriba `dgsDecision`, `dgcCurved` o `dgtSelect`.
- Los ejemplos de esta ayuda pasan todos los argumentos de un método explícitamente, por ejemplo `AddConnector(Start, Check, "", "")`, de modo que compilan con cualquier versión de los encabezados generados.
- Los eventos son closures, de modo que sus manejadores son funciones miembro __fastcall de una clase. Un parámetro var como aAccept es una referencia bool.
- TsgcDiagramRenderer es una clase simple: créela con `new TsgcDiagramRenderer()` y elimínela con delete.

En el código

Consulte también

- Inicio rápido VCL
- Plataformas

Inicio rápido sin ventana

Un diagrama no necesita una vista. TsgcDiagramDocument y TsgcDiagramRenderer usan solo la RTL y sgcUICore, de modo que una aplicación de consola, un servicio o un servidor Linux64 puede construir un diagrama, enrutarlo y escribir archivos PNG, SVG y JSON.

- Cree el documento con `TsgcDiagramDocument.Create` (sin propietario) y libérelo usted mismo.
- Los conectores se enrutan al final de cada conjunto de cambios mientras Routing.AutoRoute es True (el valor predeterminado). Llame a RouteAll para enrutar de nuevo todos los conectores.
- El renderizador exporta los límites del contenido más un margen, 20 unidades del mundo de forma predeterminada, a la escala que usted pase.

Ejemplo

```
uses
  SysUtils, Classes,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_Render;

procedure ExportOrderFlow(const aFolder: string);
var
  oDocument: TsgcDiagramDocument;
  oRenderer: TsgcDiagramRenderer;
  oStart, oPay, oShip: TsgcDiagramNode;
begin
  oDocument := TsgcDiagramDocument.Create;
  try
    oDocument.BeginUpdate;
    try
      oStart := oDocument.AddNode(dgsTerminator, 0, 0, 140, 50, 'New order');
      oPay := oDocument.AddNode(dgsProcess, 0, 100, 140, 60, 'Take payment');
      oShip := oDocument.AddNode(dgsProcess, 0, 210, 140, 60, 'Ship');
      oDocument.AddConnector(oStart, oPay);
      oDocument.AddConnector(oPay, oShip);
    finally
      oDocument.EndUpdate; // the connectors are routed here
    end;
    oRenderer := TsgcDiagramRenderer.Create;
    try
      oRenderer.SaveToPNGFile(oDocument, IncludeTrailingPathDelimiter(aFolder) + 'order.png', 2);
      oRenderer.SaveToSVGFile(oDocument, IncludeTrailingPathDelimiter(aFolder) + 'order.svg');
    finally
      oRenderer.Free;
    end;
    oDocument.SaveToFile(IncludeTrailingPathDelimiter(aFolder) + 'order.json');
  finally
    oDocument.Free;
  end;
end;
```

Consulte también

- Exportación
- Serialización
- Plataformas
- TsgcDiagramDocument
- TsgcDiagramRenderer
- Diseño por capas

TsgcVCLDiagramView

TsgcVCLDiagramView (unidad sgcDiagram_VCL) es el control VCL que muestra y edita un diagrama. Posee un TsgcDiagramDocument, publicado como Diagram, y un TsgcDiagramController que gestiona el ratón, el teclado y el arrastrar y soltar.

Propiedades

Propiedad	Significado
Diagram	El TsgcDiagramDocument que muestra y edita la vista. Se guarda en el archivo del formulario como la propiedad DiagramJSON, escrita solo cuando el diagrama no está vacío.
Tool	La herramienta de puntero: dgtSelect (predeterminado), dgtPan, dgtConnector o dgtShape.
ShapeTool	La forma que dibuja la herramienta dgtShape, dgsRectangle de forma predeterminada.
ReadOnly	True convierte la vista en un visor: la selección, el hover, el zoom y el desplazamiento funcionan, la edición no.
Zoom, MinZoom, MaxZoom	El factor de zoom y sus límites (0.05 y 8 de forma predeterminada).
ShowGrid	Dibuja la cuadrícula de Diagram.GridSize (True de forma predeterminada).
SnapToGrid, SnapToGuides	Ajusta los nodos movidos y redimensionados a la cuadrícula y a los bordes, centros e intervalos iguales de los nodos de alrededor (ambos True de forma predeterminada).
ShowPorts	Cuándo se dibujan los puertos de los nodos: dgvNever, dgvHover (predeterminado) o dgvAlways.
Theme	El tema del diagrama, el mismo valor que Diagram.Theme (no se almacena por separado).
FontFile	Un archivo de fuente que se usa para dibujar el texto, la misma fuente en la vista y en la exportación.
MaxFPS	La mayor frecuencia de repintado del control.
Controller	Pública. El TsgcDiagramController que convierte el puntero y el teclado en ediciones.

Métodos

Método	Efecto
ZoomToFit, ZoomToSelection	Aplican zoom para que todo el diagrama, o la selección, llene la vista.
CenterOn(aPoint)	Desplaza un punto del mundo al centro de la vista.
ScreenToDiagram, DiagramToScreen	Convierten entre los píxeles del dispositivo de la vista y las unidades del mundo.

Undo, Redo	Deshacen o rehacen el último paso del documento.
CopyToClipboard, CutToClipboard, PasteFromClipboard	El portapapeles del diagrama, consulte Deshacer y portapapeles.
DeleteSelection	Elimina los elementos seleccionados como un único paso de deshacer.
EditText(aNode)	Abre el editor en el lugar sobre el texto de un nodo; devuelve False cuando no se permite la edición.
SaveToPNG, SaveToSVG	Exportan a un archivo o a un stream, con una escala opcional (1 de forma predeterminada).

Eventos

La vista publica OnChange, OnSelectionChange, OnItemClick, OnItemDblClick, OnHover, OnConnecting, OnConnected, OnDrawNode, OnTextEditing, OnTextEdited y OnZoomChange, descritos en Eventos.

Notas

- La vista no tiene barras de desplazamiento. La rueda desplaza, Shift y la rueda desplazan en horizontal, Ctrl y la rueda hacen zoom alrededor del puntero, y el botón central, la barra espaciadora o la herramienta dgtPan desplazan la vista, consulte Edición y teclado.
- El editor de texto es un memo en el lugar: Enter o salir del memo confirma, Esc cancela y Shift+Enter añade una línea.
- Soltar un elemento de TsgcVCLDiagramPalette crea el nodo, o el fragmento, en el punto de soltado.

Ejemplo

```
uses
  Forms,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

procedure AddStepAtCentre(aView: TsgcVCLDiagramView);
var
  vPoint: TsgcDiagramPoint;
  oNode: TsgcDiagramNode;
begin
  // the world point under the centre of the control
  vPoint := aView.ScreenToDiagram(sgcDiagramPoint(aView.ClientWidth / 2,
    aView.ClientHeight / 2));
  oNode := aView.Diagram.AddNode(dgsProcess, vPoint.X - 60, vPoint.Y - 30,
    120, 60, 'New step');
  aView.Diagram.Selection.Clear;
  aView.Diagram.Selection.Add(oNode);
  aView.EditText(oNode);
end;

procedure ExportView(aView: TsgcVCLDiagramView);
begin
  aView.SaveToPNG('diagram.png', 2);
  aView.SaveToSVG('diagram.svg');
end;
```

Consulte también

- TsgcFMXDiagramView
- TsgcVCLDiagramPalette

- TsgcDiagramDocument
- Inicio rápido VCL

TsgcFMXDiagramView

TsgcFMXDiagramView (unidad sgcDiagram_FMX) es el control FireMonkey que muestra y edita un diagrama en Windows, macOS, Linux, iOS y Android. Tiene las mismas propiedades, métodos y eventos que TsgcVCLDiagramView: Diagram, Tool, ShapeTool, ReadOnly, Zoom, MinZoom, MaxZoom, ShowGrid, SnapToGrid, SnapToGuides, ShowPorts, Theme, FontFile, MaxFPS, Controller, los métodos de zoom, portapapeles, deshacer y exportación, y OnChange, OnSelectionChange, OnItemClick, OnItemDblClick, OnHover, OnConnecting, OnConnected, OnDrawNode, OnTextEditing, OnTextEdited y OnZoomChange, descritos en Eventos.

Diferencias con la vista VCL

- El diagrama se guarda en el archivo fmx como la propiedad DiagramJSON, escrita solo cuando el diagrama no está vacío.
- Las posiciones están en píxeles del dispositivo: las coordenadas locales del control por su escala.
- Un gesto de pellizco hace zoom alrededor del centro del pellizco. Un dedo que empieza sobre un elemento lo arrastra; un dedo que empieza en un espacio vacío desplaza la vista.
- La rueda del ratón de FMX no tiene posición, de modo que la rueda hace zoom alrededor de la última posición del puntero.
- iOS y Android no tienen servicio de arrastrar y soltar: toque un elemento de la paleta para armarlo, y luego toque la vista para colocar la forma.
- El formulario FMX gestiona Tab antes que el control con foco, de modo que Tab y Shift+Tab no recorren los elementos.
- El editor de texto es un TMemo FMX simple que sigue el estilo del formulario.

Ejemplo

```
uses
  FMX.Forms, FMX.Types,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_FMX;

function CreateTouchViewer(aForm: TForm; const aFileName: string)
  : TsgcFMXDiagramView;
begin
  Result := TsgcFMXDiagramView.Create(aForm);
  Result.Parent := aForm;
  Result.Align := TAlignLayout.Client;
  Result.ReadOnly := True;
  Result.ShowGrid := False;
  Result.ShowPorts := dgvNever;
  Result.MinZoom := 0.25;
  Result.Diagram.LoadFromFile(aFileName);
  Result.ZoomToFit;
end;
```

Consulte también

- TsgcVCLDiagramView
- TsgcFMXDiagramPalette
- Inicio rápido FMX
- Modo visor

TsgcVCLDiagramPalette

TsgcVCLDiagramPalette (unidad sgcDiagram_VCL) muestra las formas que el usuario puede añadir a un TsgcVCLDiagramView, con un clic o arrastrando y soltando.

Propiedades

Propiedad	Significado
View	La vista de diagrama que recibe los elementos en los que se hace clic o que se arrastran.
Items	Los elementos, una colección de TsgcDiagramPaletteItem mostrada en una cuadrícula, agrupada por Category.
IconSize	El tamaño del icono de un elemento en píxeles, 32 de forma predeterminada.
ShowCaptions	Dibuja la leyenda bajo cada icono (True de forma predeterminada).
Theme	utLight (predeterminado), utDark o utCustom.
MaxFPS	La mayor frecuencia de repintado del control.
Model	Pública. El modelo de la paleta detrás del control: diseño, prueba de impacto y carga de arrastre.
DragPayload	Pública. La carga del arrastre en curso: el JSON del elemento arrastrado.

Elementos

TsgcDiagramPaletteItem	Significado
Category, Caption	El encabezado del grupo y el texto bajo el icono.
Shape, Width, Height, Text, Style, PathData	El nodo que crea el elemento (PathData para dgsPath, en un cuadro de 0..100).
Fragment	Un fragmento JSON, tal como lo devuelve Diagram.CopySelection. Un elemento con fragmento suelta el fragmento entero.
Payload	De solo lectura. El JSON que suelta el elemento.

Eventos

Evento	Cuándo
OnItemClick(Sender; aItem)	Se hace clic en un elemento.

Notas

- Un clic en un elemento establece el Tool de la vista en dgtShape y su ShapeTool en la forma del elemento; un clic o un arrastre en la vista crea entonces el nodo, y Esc vuelve a dgtSelect. Un elemento de fragmento solo se coloca arrastrando y soltando.
- Arrastrar un elemento a la vista muestra una vista previa y suelta el nodo, o el fragmento, en el puntero como un único paso de deshacer. La carga nunca pasa por el portapapeles del sistema.
- Items.AddShape(aCategory, aCaption, aShape, aWidth, aHeight, aText) e Items.AddFragment(aCategory, aCaption, aFragment) añaden elementos en el código. LoadDefaultItems restaura los elementos predeterminados.

Ejemplo

```
uses
  Controls, Forms,
  sgcUICore_Theme, sgcDiagram_Types, sgcDiagram_Palette, sgcDiagram_VCL;

procedure CreateFlowPalette(aForm: TForm; aView: TsgcVCLDiagramView);
var
  oPalette: TsgcVCLDiagramPalette;
  oItem: TsgcDiagramPaletteItem;
begin
  oPalette := TsgcVCLDiagramPalette.Create(aForm);
  oPalette.Parent := aForm;
  oPalette.Align := alLeft;
  oPalette.Width := 200;
  oPalette.View := aView;
  oPalette.Items.Clear;
  oPalette.Items.AddShape('Flow', 'Step', dgsProcess, 120, 60, 'Step');
  oPalette.Items.AddShape('Flow', 'Choice', dgsDecision, 140, 80, '?');
  oItem := oPalette.Items.AddShape('Notes', 'Note', dgsText, 160, 40, 'Note');
  oItem.Style.Fill := sgcUIARGB($FF, $FF, $F5, $9D);
  // the current selection of the view as a reusable block
  if aView.Diagram.Selection.Count > 0 then
    oPalette.Items.AddFragment('Blocks', 'My block',
      aView.Diagram.CopySelection);
end;
```

Consulte también

- TsgcFMXDiagramPalette
- TsgcVCLDiagramView
- Formas

TsgcFMXDiagramPalette

TsgcFMXDiagramPalette (unidad sgcDiagram_FMX) es la paleta FireMonkey para un TsgcFMXDiagramView. Tiene las propiedades, los elementos y los eventos de TsgcVCLDiagramPalette: View, Items, IconSize, ShowCaptions, Theme, MaxFPS, Model, DragPayload, OnItemClick y LoadDefaultItems.

Diferencias con la paleta VCL

- En las plataformas de escritorio, un arrastre inicia el arrastrar y soltar FMX del formulario; la vista muestra la vista previa mientras el puntero está sobre ella.
- En iOS y Android no hay servicio de arrastre: un toque en un elemento arma la herramienta de forma de la vista, y un toque en la vista coloca la forma.

Notas

- Un clic en un elemento establece el Tool de la vista en dgtShape y su ShapeTool en la forma del elemento; un clic o un arrastre en la vista crea entonces el nodo, y Esc vuelve a dgtSelect. Un elemento de fragmento solo se coloca arrastrando y soltando.
- Arrastrar un elemento a la vista muestra una vista previa y suelta el nodo, o el fragmento, en el puntero como un único paso de deshacer. La carga nunca pasa por el portapapeles del sistema.
- Items.AddShape(aCategory, aCaption, aShape, aWidth, aHeight, aText) e Items.AddFragment(aCategory, aCaption, aFragment) añaden elementos en el código. LoadDefaultItems restaura los elementos predeterminados.

Ejemplo

```
uses
  FMX.Forms, FMX.Types,
  sgcDiagram_Types, sgcDiagram_Palette, sgcDiagram_FMX;

procedure CreateOrgPalette(aForm: TForm; aView: TsgcFMXDiagramView);
var
  oPalette: TsgcFMXDiagramPalette;
begin
  oPalette := TsgcFMXDiagramPalette.Create(aForm);
  oPalette.Parent := aForm;
  oPalette.Align := TAlignLayout.Left;
  oPalette.Width := 160;
  oPalette.IconSize := 40;
  oPalette.View := aView;
  oPalette.Items.Clear;
  oPalette.Items.AddShape('People', 'Manager', dgsRoundRect, 160, 60, 'Manager');
  oPalette.Items.AddShape('People', 'Employee', dgsRectangle, 140, 50, 'Employee');
end;
```

Consulte también

- TsgcFMXDiagramView
- TsgcVCLDiagramPalette

TsgcDiagramDataBinding

TsgcDiagramDataBinding (unidad sgcDiagram_Data) construye un diagrama a partir de filas, normalmente un organigrama o un diagrama de flujo: cada fila se convierte en un nodo, y cada clave de padre en un conector del padre al hijo. Es independiente del framework y funciona con la vista VCL, la vista FMX o un documento solo. Los nombres de campos y columnas no distinguen mayúsculas de minúsculas.

Propiedades

Propiedad	Significado
DataSource	El TDataSource cuyo DataSet proporciona las filas.
Table	Pública. Una IsgcUITable que proporciona las filas; cuando se asigna, se lee en lugar de DataSource. Establézcala en nil antes de liberar el objeto de tabla.
KeyField	El campo con la clave única de una fila. La clave se convierte en el Id del nodo, de modo que Document.FindNode(key) encuentra el nodo.
ParentField	El campo con la clave del padre, o varias claves separadas por ParentSeparator.
ParentSeparator	El separador de varias claves de padre, un punto y coma de forma predeterminada. Varios padres forman un diagrama de flujo en lugar de un árbol.
LabelField	El campo que se muestra como texto del nodo; la clave cuando está vacío.
Target	La vista que muestra el diagrama: cualquier componente que implemente IsgcDiagramDocumentOwner, como TsgcVCLDiagramView y TsgcFMXDiagramView.
Document	Pública. Un documento que se rellena directamente; cuando se asigna, se usa en lugar del documento de Target. No es de su propiedad: establézcalo en nil antes de liberar el documento.
Layout	dgbAuto (predeterminado) aplica el diseño en árbol cuando ningún nodo tiene dos padres, y si no el diseño por capas; dgbTree y dgbLayered fuerzan uno de ellos; dgbNone mantiene los nodos en filas, en el orden de las filas.
Direction, TreeStyle	Se pasan al diseño: dglTopToBottom y dgoStandard de forma predeterminada.
NodeShape, NodeWidth, NodeHeight	La forma y el tamaño de los nodos: dgsRoundRect, 140 por 60 de forma predeterminada.
ConnectorKind	El tipo de los conectores, dgcOrthogonal de forma predeterminada.
LiveUpdates	True (predeterminado) reconstruye el diagrama después de que cambien los datos.
Active	True construye el diagrama y empieza a seguir los datos.

Métodos y eventos

Miembro	Significado

Refresh	Reconstruye el diagrama ahora. Genera EsgcDiagramError dentro de una transacción del documento.
OnCreateNode(Sender; aNode; aRow)	Se ejecuta para cada nodo después de establecer su texto, forma y tamaño. aRow es la fila basada en 0, y el DataSet está posicionado en ella.
OnCreateConnector(Sender; aConnector; aRow)	Se ejecuta para cada conector una vez que existen todos los nodos. aRow es la fila del nodo hijo; el DataSet no está posicionado en ella.

Reglas

- Una fila con clave vacía, o con una clave que usó una fila anterior, no genera nodo. Una clave de padre vacía, desconocida o repetida, o la propia clave de la fila, no genera conector, de modo que una fila cuyos padres son todos desconocidos es una raíz.
- Una reconstrucción borra el documento (elementos, capas e imágenes), crea los elementos, aplica el diseño y enruta los conectores en un único BeginUpdate, de modo que la vista se repinta una vez. No es un paso de deshacer y borra las dos pilas de deshacer.
- El enlace es unidireccional: nunca escribe en los datos, y los cambios hechos en el diagrama se sustituyen en la siguiente reconstrucción.
- No se ejecuta ninguna reconstrucción en tiempo de diseño, de modo que las vistas nunca almacenan un diagrama generado en el archivo del formulario.
- Consulte Enlace de datos para las actualizaciones en vivo y un ejemplo completo.

Edición Community

TsgcDiagramDataBinding requiere SGC_DIAGRAM_ADVANCED: en la edición Community la unidad sgcDiagram_Data está vacía y el componente no se registra.

Ejemplo

```
uses
  DB,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_Data, sgcDiagram_VCL;

function BindStaff(aView: TsgcVCLDiagramView; aDataSource: TDataSource)
  : TsgcDiagramDataBinding;
begin
  Result := TsgcDiagramDataBinding.Create(aView.Owner);
  Result.KeyField := 'EmpNo';
  Result.ParentField := 'ManagerNo';
  Result.LabelField := 'FullName';
  Result.Layout := dgbTree;
  Result.TreeStyle := dgoCompactLeaves;
  Result.Target := aView;
  Result.DataSource := aDataSource;
  Result.Active := True;
end;
```

Consulte también

- Enlace de datos
- Diseño en árbol
- Diseño por capas
- Ediciones

Eventos

Los eventos de las vistas proceden de su controlador; el documento y el enlace tienen los suyos. Las vistas VCL y FMX publican los mismos eventos con los mismos parámetros.

Eventos de la vista

Evento	Cuándo
OnChange(Sender; aChanges)	El documento cambió. aChanges es un conjunto de dgkGeometry, dgkStyle, dgkStructure, dgkSelection y dgkView.
OnSelectionChange(Sender)	La selección cambió.
OnItemClick(Sender; aItem), OnItemDblClick(Sender; aItem)	Se hace clic o doble clic en un nodo o conector.
OnHover(Sender; aItem)	El puntero entra en un elemento, o sale de él (aItem es nil).
OnConnecting(Sender; aSource, aSourcePort, aTarget, aTargetPort; var aAccept)	Un arrastre de conector pregunta antes de anclar un extremo. Un puerto vacío es el contorno del nodo, un nodo nil es un extremo libre. Establezca aAccept en False para rechazar.
OnConnected(Sender; aItem)	El usuario creó o reconectó un conector.
OnDrawNode(Sender; aCanvas; aNode; aRect; aZoom; aScale; var aDefault)	Antes de pintar un nodo. aRect es el cuadro sin rotar en unidades del mundo, y la transformación del canvas asigna unidades del mundo a píxeles del dispositivo con la rotación del nodo aplicada; una unidad del mundo equivale a aZoom x aScale píxeles del dispositivo. Dibuje en aCanvas, un TsgcUICanvas, y establezca aDefault en False para omitir el pintado predeterminado de la forma, la imagen y el texto. Las leyendas de los puertos se pintan en cualquier caso.
OnTextEditing(Sender; aNode; var aAllow)	El editor de texto está a punto de abrirse. Establezca aAllow en False para mantener el texto de solo lectura.
OnTextEdited(Sender; aNode)	El editor cambió el texto de un nodo.
OnZoomChange(Sender)	El zoom cambió.

Otros eventos

Evento	Cuándo
TsgcDiagramDocument.OnChange, OnSelectionChange	Las mismas notificaciones, para un documento sin vista.
TsgcDiagramDocument.OnDecodeImage(Sender; aKey; aStream; var aImage; var aHandled)	Se carga una imagen en un formato distinto de PNG: decodifique aStream en aImage y establezca aHandled.
TsgcDiagramDataBinding.OnCreateNode, OnCreateConnector	Consulte TsgcDiagramDataBinding.

TsgcVCLDiagramPalette.OnItemClick, TsgcFMXDiagramPalette.OnItemClick	Se hace clic en un elemento de la paleta.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

type
  TDiagramRules = class
  private
    FDirty: Boolean;
    procedure DoConnecting(Sender: TObject; aSource: TsgcDiagramNode;
      const aSourcePort: string; aTarget: TsgcDiagramNode;
      const aTargetPort: string; var aAccept: Boolean);
    procedure DoTextEditing(Sender: TObject; aNode: TsgcDiagramNode;
      var aAllow: Boolean);
    procedure DoChange(Sender: TObject; aChanges: TsgcDiagramChanges);
  public
    procedure Attach(aView: TsgcVCLDiagramView);
    property Dirty: Boolean read FDirty write FDirty;
  end;

procedure TDiagramRules.Attach(aView: TsgcVCLDiagramView);
begin
  aView.OnConnecting := DoConnecting;
  aView.OnTextEditing := DoTextEditing;
  aView.OnChange := DoChange;
end;

procedure TDiagramRules.DoConnecting(Sender: TObject; aSource: TsgcDiagramNode;
  const aSourcePort: string; aTarget: TsgcDiagramNode;
  const aTargetPort: string; var aAccept: Boolean);
begin
  // no loops, and nothing goes back into a terminator
  if aTarget <> nil then
    aAccept := (aTarget <> aSource) and (aTarget.Shape <> dgsTerminator);
end;

procedure TDiagramRules.DoTextEditing(Sender: TObject; aNode: TsgcDiagramNode;
  var aAllow: Boolean);
begin
  aAllow := not aNode.Locked;
end;

procedure TDiagramRules.DoChange(Sender: TObject; aChanges: TsgcDiagramChanges);
begin
  if aChanges * [dgkGeometry, dgkStyle, dgkStructure] <> [] then
    FDirty := True;
end;
```

Consulte también

- TsgcVCLDiagramView
- TsgcDiagramController
- Puertos y anclaje

TsgcDiagramDocument

TsgcDiagramDocument (unidad sgcDiagram_Model) contiene un diagrama: sus nodos, conectores, capas e imágenes, la selección y el historial de deshacer. Cada vista muestra un documento, publicado como su propiedad Diagram, y un documento también funciona solo, sin vista, en una aplicación de consola o en un servicio.

Propiedades

Propiedad	Significado
Nodes, Connectors	Los elementos, en orden z (el último queda encima).
Layers, Images	Las capas y las imágenes que muestran los nodos de imagen, por clave (Images.Add, AddPNG, Find, Delete).
Selection	Los elementos seleccionados: Add, Remove, Toggle, Clear, SelectAll, Contains, Count e Items.
Theme	utLight (predeterminado), utDark o utCustom.
GridSize	El paso de la cuadrícula en unidades del mundo, 10 de forma predeterminada.
NodeStyle, ConnectorStyle	Los estilos usados cuando un elemento deja un valor en 0.
DefaultConnectorKind	El tipo de los nuevos conectores, dgcOrthogonal de forma predeterminada.
Routing	Los ajustes del enrutador: Margin (12), BendPenalty (40), Spacing (6) y AutoRoute (True).
UndoLimit	El número de pasos de deshacer que se conservan, 100 de forma predeterminada.
InTransaction, Loading	Indicadores de estado de solo lectura.

Métodos

Método	Efecto
AddNode(aShape, aX, aY, aWidth, aHeight, aText)	Añade un nodo y lo devuelve.
AddConnector(aSource, aTarget, aSourcePort, aTargetPort)	Añade un conector entre dos nodos; un puerto vacío ancla el extremo al contorno.
FindNode(aId), FindConnector(aId)	Encuentran un elemento por Id, nil cuando no se encuentra.
ConnectorsOf(aNode, aList), ChildrenOf(aContainer, aList)	Rellenan una TList con los conectores de un nodo, o los hijos de un contenedor.
ItemsInRect(aRect, aList, aFullyInside)	Rellena una TList con los elementos que tocan, o quedan dentro de, un rectángulo.
ContentBounds	Los límites de todos los elementos visibles.

DeleteSelection, Clear	Eliminan los elementos seleccionados, o todo.
Group, Ungroup, BringToFront, SendToBack	Actúan sobre la selección. Group devuelve el nuevo Id del grupo.
BeginUpdate, EndUpdate	Un repintado y una notificación de cambio para muchos cambios.
BeginTransaction(aCaption), EndTransaction, Rollback	Muchos cambios como un único paso de deshacer, o ninguno de ellos.
BeginUndoSuspend, EndUndoSuspend	Los cambios intermedios no se registran y borran las dos pilas de deshacer; para cargas masivas.
Undo, Redo, CanUndo, CanRedo, UndoCaption, RedoCaption	El historial de deshacer.
RouteAll	Enruta de nuevo todos los conectores.
SaveToJSON, LoadFromJSON, SaveToFile, LoadFromFile, SaveToStream, LoadFromStream	El formato JSON.
CopySelection, Paste(aJSON)	Un fragmento del documento como JSON, y su pegado.
ResolveNodeStyle(aNode), ResolveConnectorStyle(aConnector)	El estilo con el que se dibuja un elemento, con los colores del tema resueltos.
LayerVisible(aName), LayerLocked(aName)	Cómo se trata una capa.

Eventos

OnChange(Sender; aChanges), OnSelectionChange y OnDecodeImage, consulte Eventos.

Hilos

Un documento solo se modifica en el hilo que lo posee, el hilo principal cuando se muestra en una vista.

Ejemplo

```
uses
  Classes, SysUtils,
  sgcDiagram_Types, sgcDiagram_Model;

procedure ImportGrid(aDocument: TsgcDiagramDocument; aCount: Integer);
var
  i: Integer;
begin
  // a bulk load: no undo history, one repaint
  aDocument.BeginUndoSuspend;
  aDocument.BeginUpdate;
  try
    for i := 0 to aCount - 1 do
      aDocument.AddNode(dgsRectangle, (i mod 10) * 160, (i div 10) * 100,
        120, 60, 'Item ' + IntToStr(i + 1));
  finally
    aDocument.EndUpdate;
    aDocument.EndUndoSuspend;
  end;
end;
```

```
function CountIncoming(aDocument: TsgcDiagramDocument;
  const aId: string): Integer;
var
  oNode: TsgcDiagramNode;
  oList: TList;
  i: Integer;
begin
  Result := 0;
  oNode := aDocument.FindNode(aId);
  if oNode = nil then
    Exit;
  oList := TList.Create;
  try
    aDocument.ConnectorsOf(oNode, oList);
    for i := 0 to oList.Count - 1 do
      if TsgcDiagramConnector(oList[i]).TargetNodeId = oNode.Id then
        Inc(Result);
  finally
    oList.Free;
  end;
end;
```

Consulte también

- TsgcDiagramNode
- TsgcDiagramConnector
- Deshacer y portapapeles
- Arquitectura

TsgcDiagramNode

Un nodo es una forma con un texto: un paso de proceso, una decisión, una persona de un organigrama o un contenedor de otros nodos. Los nodos se crean con Document.AddNode y viven en Document.Nodes.

Propiedades

Propiedad	Significado
Shape	La forma, dgsRectangle de forma predeterminada.
Left, Top, Width, Height	El cuadro en unidades del mundo (120 por 60 de forma predeterminada).
Angle	La rotación en grados, en sentido horario, alrededor del centro.
Text, TextAlign, TextVAlign	El texto y su alineación (centrado de forma predeterminada).
Ports	Los puertos; vacío usa los puertos predeterminados de la forma.
ContainerId	El Id del nodo contenedor que contiene este nodo.
GroupId	El grupo del nodo; los nodos de un grupo se seleccionan y se mueven juntos.
ImageKey	La clave en Document.Images de la imagen que dibuja un nodo dgsImage.
PathData	El contorno de un nodo dgsPath, en sintaxis de ruta SVG en un cuadro de 0..100.
Visible	False oculta el nodo.
Id	El Id único del elemento en el documento; AddNode y AddConnector generan uno.
Style	El estilo del elemento; los valores en 0 provienen del estilo del documento.
Layer	El nombre de la capa; vacío para la capa predeterminada.
Locked	True impide que el elemento se seleccione y se edite.
Data, Tag	Datos de la aplicación: una cadena (guardada en el archivo) y un entero.
Selected, Document	Pública, de solo lectura.

Métodos

Método	Efecto
SetBounds(aLeft, aTop, aWidth, aHeight), MoveBy(aDX, aDY)	Mueven o redimensionan el nodo en un único cambio.
BoundsRect, Center	El cuadro y su centro.
PortPoint(aPortId)	La posición del mundo de un puerto.

Ejemplo

```
uses
  SysUtils,
  sgcUICore_Canvas, sgcDiagram_Types, sgcDiagram_Model;

function AddOrderNode(aDocument: TsgcDiagramDocument;
  aOrderNo: Integer): TsgcDiagramNode;
begin
  Result := aDocument.AddNode(dgsDocument, 40, 40, 160, 80, 'Order');
  Result.TextAlign := thaLeft;
  Result.Tag := aOrderNo;
  Result.Data := '{"order":' + IntToStr(aOrderNo) + '}';
  Result.Angle := -5;
end;

procedure PutInContainer(aContainer, aNode: TsgcDiagramNode);
begin
  aNode.ContainerId := aContainer.Id;
  aNode.SetBounds(aContainer.Left + 20, aContainer.Top + 40, aNode.Width,
    aNode.Height);
end;
```

Consulte también

- TsgcDiagramPort
- TsgcDiagramConnector
- Formas

TsgcDiagramConnector

Un conector une dos nodos, o un nodo y un punto libre. Su ruta la calcula el enrutador a partir de su Kind. Los conectores se crean con Document.AddConnector y viven en Document.Connectors.

Propiedades

Propiedad	Significado
Kind	dgcStraight, dgcOrthogonal o dgcCurved.
SourceNodeId, SourcePortId, TargetNodeId, TargetPortId	Los extremos anclados. Un puerto vacío ancla el extremo al contorno del nodo.
SourcePoint, TargetPoint	Los extremos libres, usados cuando el Id del nodo está vacío.
Waypoints	Los puntos por los que pasa un conector recto o curvo.
StartArrow, EndArrow, ArrowSize	Las puntas de flecha (dgaNone y dgaTriangle de forma predeterminada) y su tamaño (10).
Labels	Las etiquetas: Text, Position a lo largo de la ruta de 0 a 1 (0.5 de forma predeterminada) y Offset respecto a la ruta.
Points	Pública, de solo lectura. La ruta actual.
RouteFailed	Pública, de solo lectura. True cuando el enrutador no encontró ningún camino alrededor de los nodos.
Id	El Id único del elemento en el documento; AddNode y AddConnector generan uno.
Style	El estilo del elemento; los valores en 0 provienen del estilo del documento.
Layer	El nombre de la capa; vacío para la capa predeterminada.
Locked	True impide que el elemento se seleccione y se edite.
Data, Tag	Datos de la aplicación: una cadena (guardada en el archivo) y un entero.
Selected, Document	Pública, de solo lectura.

Métodos

Método	Efecto
SetRoute(aPoints, aFailed)	Establece la ruta; lo usan el enrutador y los diseños.

Edición Community

Sin SGC_DIAGRAM_ADVANCED, un conector dgcOrthogonal se enruta como un codo: sin esquiva de obstáculos y sin separación, y RouteFailed permanece en False.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model;

procedure ConnectWithLabel(aDocument: TsgcDiagramDocument;
  aFrom, aTo: TsgcDiagramNode; const aText: string);
var
  oConnector: TsgcDiagramConnector;
begin
  oConnector := aDocument.AddConnector(aFrom, aTo);
  oConnector.Kind := dgcCurved;
  oConnector.StartArrow := dgaCircle;
  oConnector.EndArrow := dgaOpenTriangle;
  oConnector.Style.Dash := dgdDash;
  with oConnector.Labels.Add do
  begin
    Text := aText;
    Position := 0.25;
    Offset := -12;
  end;
end;
```

Consulte también

- Conectores y enrutamiento
- TsgcDiagramPort
- TsgcDiagramNode

TsgcDiagramPort

Un puerto es un punto con nombre en un nodo donde se anclan los extremos de los conectores. Un nodo con una colección Ports vacía usa los puertos predeterminados de su forma: los puntos medios de los lados, llamados left, top, right y bottom.

Propiedades

Propiedad	Significado
Id	El nombre que usan SourcePortId y TargetPortId.
X, Y	La posición como fracción del cuadro del nodo, de 0 a 1 (0.5 de forma predeterminada).
Side	El lado por el que sale un conector: dgpAuto (predeterminado), dgpLeft, dgpTop, dgpRight o dgpBottom.
Flow	dgfBoth (predeterminado), dgfIn o dgfOut: qué extremos pueden anclarse al puerto.
Caption	Un texto dibujado junto al puerto.

Node.Ports.Add añade un puerto y Node.Ports.Find(aId) encuentra uno. Node.PortPoint(aId) devuelve su posición del mundo.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model;

procedure AddTypedPorts(aNode: TsgcDiagramNode);
begin
  with aNode.Ports.Add do
  begin
    Id := 'in';
    X := 0;
    Y := 0.5;
    Side := dgpLeft;
    Flow := dgfIn;
    Caption := 'Input';
  end;
  with aNode.Ports.Add do
  begin
    Id := 'out';
    X := 1;
    Y := 0.5;
    Side := dgpRight;
    Flow := dgfOut;
    Caption := 'Output';
  end;
end;
```

Consulte también

- Puertos y anclaje
- TsgcDiagramNode

TsgcDiagramStyle

Cada nodo y conector tiene un Style, y el documento tiene un NodeStyle y un ConnectorStyle. Un color o un tamaño dejado en 0 proviene del estilo del documento, y un color que sigue en 0 allí proviene del tema, de modo que un diagrama sigue un cambio de tema a menos que se definan sus colores.

Propiedades

Propiedad	Significado
Fill	El color de relleno, un TsgcUIColor (cree uno con sgcUIARGB).
Stroke	El color del contorno o de la línea.
StrokeWidth	El ancho del contorno o de la línea en unidades del mundo, 1 de forma predeterminada.
Dash	dgdSolid (predeterminado), dgdDash o dgdDot.
CornerRadius	El radio de las esquinas de rectángulos y rectángulos redondeados.
Text	El estilo del texto: Size, Bold y Color.

Document.ResolveNodeStyle(aNode) y ResolveConnectorStyle(aConnector) devuelven los valores con los que se dibuja un elemento.

Ejemplo

```
uses
  sgcUICore_Theme, sgcDiagram_Types, sgcDiagram_Model;

procedure ApplyHouseStyle(aDocument: TsgcDiagramDocument);
begin
  aDocument.NodeStyle.Fill := sgcUIARGB($FF, $E3, $F2, $FD);
  aDocument.NodeStyle.Stroke := sgcUIARGB($FF, $15, $65, $C0);
  aDocument.NodeStyle.CornerRadius := 6;
  aDocument.NodeStyle.Text.Bold := True;
  aDocument.ConnectorStyle.StrokeWidth := 2;
end;

procedure MarkAsError(aNode: TsgcDiagramNode);
begin
  aNode.Style.Fill := sgcUIARGB($FF, $E5, $39, $35);
  aNode.Style.Text.Color := sgcUIARGB($FF, $FF, $FF, $FF);
end;
```

Consulte también

- Temas
- TsgcDiagramDocument

TsgcDiagramLayer

Una capa agrupa elementos por nombre. Document.Layers contiene las capas; cada nodo y conector indica su capa en la propiedad Layer, y un nombre vacío es la capa predeterminada.

Propiedades

Propiedad	Significado
Name	El nombre que usa la propiedad Layer de los elementos.
Visible	False oculta los elementos de la capa en la vista y en la exportación.
Locked	True impide que los elementos de la capa se seleccionen y se editen.

Layers.Add añade una capa y Layers.Find(aName) encuentra una. Añadir, eliminar y renombrar una capa son pasos de deshacer; Visible y Locked no lo son.

Edición Community

Sin SGC_DIAGRAM_ADVANCED, un segundo Layers.Add genera EsgcDiagramError. Un archivo con más capas se carga y se guarda sin cambios, pero cada capa se dibuja y se edita como visible y desbloqueada.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model;

procedure ToggleLayer(aDocument: TsgcDiagramDocument; const aName: string);
var
  oLayer: TsgcDiagramLayer;
begin
  oLayer := aDocument.Layers.Find(aName);
  if oLayer <> nil then
    oLayer.Visible := not oLayer.Visible;
end;
```

Consulte también

- Capas
- Ediciones

TsgcDiagramController

TsgcDiagramController (unidad sgcDiagram_Controller) es el motor de edición independiente del framework de las vistas: convierte la entrada de puntero, teclado, rueda y arrastrar y soltar en cambios del documento, mantiene el zoom y la posición de desplazamiento, y dibuja mediante un TsgcDiagramPainter. Cada vista crea uno y lo expone como su propiedad Controller. Las posiciones están en píxeles del dispositivo; internamente el controlador trabaja en unidades del mundo.

Propiedades

Propiedad	Significado
Document, Host, Painter	El documento, el control anfitrión (IsgcDiagramHost) y el painter.
Tool, ShapeTool, ShapeWidth, ShapeHeight	La herramienta activa y la forma y el tamaño predeterminado que dibuja la herramienta de forma.
ReadOnly, ShowGrid, SnapToGrid, SnapToGuides, ShowPorts	Las opciones publicadas por las vistas.
Zoom, MinZoom, MaxZoom, OffsetX, OffsetY, Viewport	La transformación de la vista.
State	La interacción en curso: dgiIdle, dgiMoving, dgiResizing y así sucesivamente.
HoverItem, EditingNode	El elemento bajo el puntero y el nodo cuyo texto se está editando.

Métodos

Método	Efecto
ZoomAt(aZoom, aX, aY), ScrollBy(aDX, aDY)	Zoom alrededor de un punto del dispositivo, desplazamiento en píxeles del dispositivo.
ZoomToFit, ZoomToSelection, CenterOn	Como en las vistas.
SelectAll, Duplicate, Nudge(aDX, aDY)	Selecciona todo, duplica la selección, la mueve en unidades del mundo.
GroupSelection, UngroupSelection	Agrupa o desagrupa los nodos seleccionados.
Undo, Redo, CopyToClipboard, CutToClipboard, PasteFromClipboard, DeleteSelection	Los comandos de edición.
EditText(aNode), EndTextEdit	Abre y cierra el editor de texto.
Cancel	Cancela la interacción en curso, como hace Esc.

Los eventos del controlador los publican las vistas, consulte Eventos. El teclado se describe en Edición y teclado.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_VCL;

procedure DuplicateAndShift(aView: TsgcVCLDiagramView);
begin
  if aView.Diagram.Selection.Count = 0 then
    Exit;
  aView.Controller.Duplicate;
  aView.Controller.Nudge(aView.Diagram.GridSize, aView.Diagram.GridSize);
end;

procedure SelectAndGroup(aView: TsgcVCLDiagramView);
begin
  aView.Controller.SelectAll;
  aView.Controller.GroupSelection;
end;
```

Consulte también

- TsgcVCLDiagramView
- Edición y teclado

TsgcDiagramRenderer

TsgcDiagramRenderer (unidad sgcDiagram_Render) dibuja un documento sin una vista: en archivos o streams PNG y SVG, o en cualquier TsgcUICanvas. Es lo que usan las vistas para SaveToPNG y SaveToSVG, y funciona en aplicaciones de consola, en servicios y en Linux.

Métodos

Método	Efecto
SaveToPNGFile(aDocument, aFileName, aScale, aMargin), SaveToPNGStream	Un PNG de todo el diagrama. aScale es 1 de forma predeterminada, aMargin 20 unidades del mundo.
SaveToSVGFile(aDocument, aFileName, aScale, aMargin), SaveToSVGStream	Lo mismo como SVG.
ExportViewport(aDocument, aScale, aMargin)	El viewport que usa una exportación: desplazamiento, zoom, escala y tamaño de la imagen.
RenderTo(aCanvas, aDocument, aViewport)	Dibuja el documento en un canvas mediante un viewport.
Painter	El TsgcDiagramPainter que dibuja; su evento OnDrawNode personaliza también los nodos en la exportación.

Ejemplo

```
uses
  sgcDiagram_Model, sgcDiagram_Render;

procedure ExportDiagram(aDocument: TsgcDiagramDocument);
var
  oRenderer: TsgcDiagramRenderer;
begin
  oRenderer := TsgcDiagramRenderer.Create;
  try
    oRenderer.SaveToPNGFile(aDocument, 'diagram.png');
    oRenderer.SaveToPNGFile(aDocument, 'diagram@2x.png', 2);
    oRenderer.SaveToSVGFile(aDocument, 'diagram.svg', 1, 40);
  finally
    oRenderer.Free;
  end;
end;
```

Consulte también

- Exportación
- Inicio rápido sin ventana

Formato JSON

SaveToJSON escribe un único objeto JSON. La salida es canónica: un orden de claves fijo, sin sangría, y todo valor igual al predeterminado se omite, de modo que una clave ausente siempre significa el valor predeterminado. Las enumeraciones se escriben como nombres, los colores como cadenas hexadecimales, los puntos como arrays [x, y]. El lector ignora las claves y los nombres desconocidos.

Documento

Clave	Valor
`format`	Siempre "sgcDiagram".
`version`	La versión del formato, 1. Un archivo de una versión más nueva genera EsgcDiagramError.
`gridSize`	GridSize (10).
`theme`	"light" (predeterminado), "dark" o "custom".
`layers`	Array de { name, visible (true), locked (false) }.
`nodes`	Array de nodos, en orden z.
`connectors`	Array de conectores, en orden z.
`images`	Array de { key, png }, los datos PNG en Base64.

NodeStyle, ConnectorStyle, DefaultConnectorKind, Routing y UndoLimit son propiedades del objeto de documento y no se escriben en el archivo.

Nodo

Clave	Propiedad
`id`	Id
`shape`	Shape: "rectangle" (default), "roundRect", "ellipse", "text", "image", "path", "process", "decision", "terminator", "data", "document", "predefinedProcess", "connectorCircle", "container"
`x` , `y` , `w` , `h`	Left, Top, Width (120), Height (60)
`angle`	Angle
`text`	Text
`textAlign` , `textVAlign`	TextAlign ("center"), TextVAlign ("middle")
`layer` , `container` , `group`	Layer, ContainerId, GroupId
`image` , `path`	ImageKey, PathData

`locked`, `visible`	Locked (false), Visible (true)
`data`, `tag`	Data, Tag
`style`	Style, véase abajo
`ports`	Array of { id, x (0.5), y (0.5), side ("auto"), flow ("both"), caption }

Conector

Clave	Propiedad
`id`	Id
`kind`	Kind: "straight", "orthogonal" (predeterminado) o "curved"
`source`, `target`	{ node, port }: SourceNodeId y SourcePortId, TargetNodeId y TargetPortId
`sourcePoint`, `targetPoint`	SourcePoint, TargetPoint de los extremos libres
`waypoints`	Waypoints
`startArrow`, `endArrow`	StartArrow ("none"), EndArrow ("triangle"): "none", "triangle", "openTriangle", "diamond", "openDiamond", "circle"
`arrowSize`	ArrowSize (10)
`labels`	Array of { text, pos (0.5), offset (0) }
`layer`, `locked`, `data`, `tag`	Layer, Locked, Data, Tag
`style`	Style, véase abajo
`points`	La ruta almacenada, que se mantiene al cargar

Style

Clave	Propiedad
`fill`, `stroke`	Fill, Stroke
`strokeWidth`	StrokeWidth (1)
`dash`	Dash: "solid" (predeterminado), "dash" o "dot"
`radius`	CornerRadius
`text`	{ size, bold, color }: el estilo de Text

Un archivo pequeño completo se muestra en Serialización.

Ejemplo

```
uses
  SysUtils,
  sgcDiagram_Types, sgcDiagram_Model;

function DiagramFromJSON(const aJSON: string): TsgcDiagramDocument;
begin
  Result := TsgcDiagramDocument.Create;
  try
    Result.LoadFromJSON(aJSON);
  except
    on E: EsgcDiagramError do
    begin
      Result.Free;
      raise Exception.Create('Not a valid diagram: ' + E.Message);
    end;
  end;
end;
```

Consulte también

- Serialización
- TsgcDiagramDocument

Diseño por capas

TsgcDiagramLayeredLayout (unidad sgcDiagram_Layout) organiza un grafo dirigido en capas, el método de Sugiyama: diagramas de flujo, diagramas de proceso, máquinas de estados y grafos de dependencias. Los conectores van de una capa a la siguiente, y el diseño mantiene bajos sus cruces.

Propiedades

Propiedad	Significado
Direction	dglTopToBottom (predeterminado), dglLeftToRight, dglBottomToTop o dglRightToLeft.
LayerSpacing	El espacio entre capas, 60 unidades del mundo de forma predeterminada, sumado al nodo más profundo de la capa.
NodeSpacing	El espacio entre nodos vecinos, 30 de forma predeterminada.
SelectionOnly	Organiza solo los nodos seleccionados.
Crossings	De solo lectura. Los cruces de conectores contados tras Execute.

Cómo funciona

- Los ciclos se rompen invirtiendo las aristas de retroceso de una búsqueda en profundidad en el orden de la colección.
- Cada nodo recibe la capa de su camino más largo desde una fuente; un conector largo recibe un nodo ficticio en cada capa que cruza.
- Los barridos por baricentro reducen los cruces y se conserva el mejor orden.
- Las coordenadas se colocan hacia la mediana de los vecinos, con conectores largos rectos, sin solapamientos.
- Las posiciones ficticias se convierten en los waypoints de los conectores rectos y curvos; los conectores ortogonales los enruta el enrutador.
- Las partes no conectadas se organizan una junto a otra, separadas por NodeSpacing.

Notas

- El diseño actúa sobre los nodos de nivel superior, o sobre los nodos seleccionados cuando SelectionOnly es True (por ejemplo, los hijos de un contenedor). Un contenedor cuenta como un nodo de su tamaño y lleva a sus hijos.
- Los nodos conservan la esquina superior izquierda de sus límites. El resultado se aplica en una transacción, de modo que un diseño es un paso de deshacer, y los conectores movidos se enrutan de nuevo, también cuando Routing.AutoRoute es False.
- Cada coordenada se calcula en centésimas enteras y toda ordenación es estable, de modo que un diseño da el mismo resultado en Win32, Win64 y Linux64.

Edición Community

Los diseños requieren SGC_DIAGRAM_ADVANCED: en la edición Community la unidad sgcDiagram_Layout está vacía.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_Layout;

function ArrangeFlowchart(aDocument: TsgcDiagramDocument): Integer;
var
  oLayout: TsgcDiagramLayeredLayout;
begin
  oLayout := TsgcDiagramLayeredLayout.Create(aDocument);
  try
    oLayout.Direction := dglLeftToRight;
    oLayout.LayerSpacing := 80;
    oLayout.NodeSpacing := 40;
    oLayout.Execute;
    Result := oLayout.Crossings;
  finally
    oLayout.Free;
  end;
end;
```

Consulte también

- Diseño en árbol
- TsgcDiagramDataBinding
- Ediciones

Diseño en árbol

TsgcDiagramTreeLayout (unidad sgcDiagram_Layout) organiza jerarquías: organigramas, árboles de categorías y mapas mentales. Los padres se centran sobre sus hijos y los subárboles vecinos se empaquetan a NodeSpacing de distancia, en tiempo lineal.

Propiedades

Propiedad	Significado
Direction	dglTopToBottom (predeterminado), dglLeftToRight, dglBottomToTop o dglRightToLeft.
LayerSpacing	El espacio entre niveles, 60 unidades del mundo de forma predeterminada, sumado al nodo más profundo del nivel.
NodeSpacing	El espacio entre subárboles vecinos, y entre hojas apiladas, 30 de forma predeterminada.
TreeStyle	dgoStandard (predeterminado) o dgoCompactLeaves.
SelectionOnly	Organiza solo los nodos seleccionados.

Cómo funciona

- Las raíces son los nodos sin conector de entrada, en el orden de la colección. Un conector va del padre al hijo; un conector que no es una arista del árbol se dibuja, pero se ignora en la colocación.
- Los hijos de un nodo conservan el orden de la colección.
- dgoCompactLeaves apila los hijos de un nodo en una columna a la derecha del centro del padre cuando tiene dos o más hijos y todos son hojas, el organigrama clásico. Los conectores rectos y curvos reciben un waypoint en codo bajo el padre.
- Varios árboles se colocan uno junto a otro en el orden de las raíces, separados por NodeSpacing.

Notas

- El diseño actúa sobre los nodos de nivel superior, o sobre los nodos seleccionados cuando SelectionOnly es True (por ejemplo, los hijos de un contenedor). Un contenedor cuenta como un nodo de su tamaño y lleva a sus hijos.
- Los nodos conservan la esquina superior izquierda de sus límites. El resultado se aplica en una transacción, de modo que un diseño es un paso de deshacer, y los conectores movidos se enrutan de nuevo, también cuando Routing.AutoRoute es False.
- Cada coordenada se calcula en centésimas enteras y toda ordenación es estable, de modo que un diseño da el mismo resultado en Win32, Win64 y Linux64.

Edición Community

Los diseños requieren SGC_DIAGRAM_ADVANCED: en la edición Community la unidad sgcDiagram_Layout está vacía.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_Layout;

procedure ArrangeOrgChart(aDocument: TsgcDiagramDocument; aCompact: Boolean);
var
  oLayout: TsgcDiagramTreeLayout;
begin
  oLayout := TsgcDiagramTreeLayout.Create(aDocument);
  try
    if aCompact then
      oLayout.TreeStyle := dgoCompactLeaves;
    oLayout.NodeSpacing := 20;
    oLayout.Execute;
  finally
    oLayout.Free;
  end;
end;
```

Consulte también

- Diseño por capas
- Enlace de datos

Enlace de datos

Un TsgcDiagramDataBinding convierte las filas de un dataset en un diagrama y lo mantiene actualizado. Esta página muestra, paso a paso, la configuración habitual de un organigrama.

Pasos

1. Dé a los datos un campo de clave y un campo de padre: cada fila es un nodo, y la clave del padre da el conector del padre. Las filas sin un padre conocido son raíces.
2. Suelte un TsgcDiagramDataBinding en el formulario, establezca DataSource, KeyField, ParentField y LabelField, y establezca Target en la vista del diagrama.
3. Elija el diseño: dgbAuto elige el diseño en árbol para un árbol y el diseño por capas cuando una fila tiene varios padres (claves separadas por ParentSeparator). Establezca Direction y TreeStyle según sea necesario.
4. Dé estilo a los nodos en OnCreateNode: el DataSet está posicionado en la fila del nodo, de modo que cualquier campo puede establecer el texto, los colores o el Data del nodo.
5. Establezca Active en True. El diagrama se construye de inmediato y se reconstruye cuando cambian los datos.

Actualizaciones en vivo

- Con LiveUpdates en True, abrir el DataSet, una edición, inserción o eliminación confirmada, un cambio en el número de filas o en el filtro, o una notificación de la tabla reconstruye el diagrama. Las solicitudes se agrupan en una reconstrucción por vuelta del bucle de mensajes.
- La navegación no reconstruye, de modo que la selección la sobrevive. Un refresh que cambia solo filas distintas de la actual, y no el número de filas, no se detecta: llame a Refresh después.
- Una reconstrucción espera mientras se edita el DataSet y mientras el usuario arrastra en la vista. Cerrar el DataSet mantiene el diagrama.
- En Delphi 10.2 y posteriores, la reconstrucción se pone en cola con TThread.ForceQueue: las aplicaciones de consola y los servicios deben llamar a CheckSynchronize.
- Una reconstrucción borra el historial de deshacer: los cambios hechos en el diagrama no se escriben de vuelta en los datos.

Edición Community

El enlace de datos requiere SGC_DIAGRAM_ADVANCED. En la edición Community, construya el diagrama en el código con AddNode y AddConnector.

Ejemplo

```
uses
  DB, SysUtils,
  sgcUICore_Theme, sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_Data,
  sgcDiagram_VCL;

type
  TOrgChart = class
  private
    FBinding: TsgcDiagramDataBinding;
    procedure DoCreateNode(Sender: TObject; aNode: TsgcDiagramNode;
      aRow: Integer);
  public
    constructor Create(aView: TsgcVCLDiagramView; aDataSource: TDataSource);
    destructor Destroy; override;
  end;
```

```
constructor TOrgChart.Create(aView: TsgcVCLDiagramView;
  aDataSource: TDataSource);
begin
  inherited Create;
  FBinding := TsgcDiagramDataBinding.Create(nil);
  FBinding.KeyField := 'EmpNo';
  FBinding.ParentField := 'ManagerNo';
  FBinding.LabelField := 'FullName';
  FBinding.Layout := dgbTree;
  FBinding.TreeStyle := dgoCompactLeaves;
  FBinding.NodeWidth := 180;
  FBinding.OnCreateNode := DoCreateNode;
  FBinding.Target := aView;
  FBinding.DataSource := aDataSource;
  FBinding.Active := True;
end;

destructor TOrgChart.Destroy;
begin
  FBinding.Free;
  inherited;
end;

procedure TOrgChart.DoCreateNode(Sender: TObject; aNode: TsgcDiagramNode;
  aRow: Integer);
var
  oDataSet: TDataSet;
begin
  oDataSet := FBinding.DataSource.DataSet;
  aNode.Text := oDataSet.FieldByName('FullName').AsString + #13#10 +
    oDataSet.FieldByName('JobTitle').AsString;
  if SameText(oDataSet.FieldByName('Dept').AsString, 'Sales') then
    aNode.Style.Fill := sgcUIARGB($FF, $C8, $E6, $C9);
end;
```

Consulte también

- TsgcDiagramDataBinding
- Diseño en árbol
- Diseño por capas

Formas

Cada nodo tiene una Shape del tipo TsgcDiagramShape. La forma decide el contorno, el área que recibe el texto, los puertos predeterminados y los colores del tema. Un nodo se coloca con Left, Top, Width y Height en unidades del mundo y se gira con Angle, en grados en sentido horario alrededor de su centro.

Forma	Uso
dgsRectangle, dgsRoundRect, dgsEllipse	Cuadros básicos. dgsRoundRect usa un radio de esquina de 8 cuando Style.CornerRadius es 0; los demás cuadros solo redondean sus esquinas cuando CornerRadius es mayor que 0.
dgsProcess, dgsDecision, dgsTerminator, dgsData, dgsDocument, dgsPredefinedProcess, dgsConnectorCircle	Los símbolos de diagrama de flujo. Un terminador siempre se redondea por la mitad de su lado menor; una decisión es un rombo cuyos puertos predeterminados son sus esquinas.
dgsText	Texto libre sin relleno ni borde.
dgsImage	Una imagen del documento, nombrada por ImageKey. Véase abajo.
dgsPath	Un contorno personalizado a partir de PathData, en sintaxis de ruta SVG dentro de un cuadro de 0 a 100 que se estira hasta el nodo.
dgsContainer	Un cuadro con una banda de encabezado que contiene otros nodos. Un nodo pertenece a él mediante ContainerId, y mover el contenedor mueve a sus hijos.

Texto

El texto se dibuja dentro del rectángulo de texto de la forma, con ajuste de línea, con TextAlign (thaCenter de forma predeterminada) y TextVAlign (tvaMiddle de forma predeterminada). El tamaño, el grosor y el color de la fuente provienen de Style.Text. En la vista, un doble clic, F2 o un carácter escrito edita el texto en el lugar.

Imágenes

Images.AddPNG(aKey, aStream) almacena los bytes PNG de un stream y los decodifica; Images.Add(aKey, aImage) añade un TsgcUIImage decodificado. Un nodo con Shape dgsImage e ImageKey establecido en la clave dibuja la imagen estirada hasta su cuadro. Otros formatos los decodifica su manejador del evento OnDecodeImage del documento. Eliminar una imagen que un nodo aún usa genera EsgcDiagramError.

Grupos y contenedores

Document.Group coloca los nodos seleccionados en un grupo y devuelve su GroupId; un clic selecciona entonces el grupo entero, y Alt con un clic selecciona un miembro. Ungroup elimina el grupo de la selección. Un contenedor posee a sus hijos: se mueven con él, un contenedor eliminado los elimina, y un nodo soltado sobre un contenedor en la vista se convierte en su hijo.

Ejemplo

```
uses
  Classes,
  sgcDiagram_Types, sgcDiagram_Model;
```

```
procedure AddShapes(aDocument: TsgcDiagramDocument; aLogo: TStream);
var
  oBox, oStar, oLogo, oChild: TsgcDiagramNode;
begin
  oBox := aDocument.AddNode(dgsContainer, 0, 0, 400, 260, 'Warehouse');
  oChild := aDocument.AddNode(dgsProcess, 20, 60, 140, 60, 'Pick');
  oChild.ContainerId := oBox.Id;
  // a five pointed star in the 0..100 path box
  oStar := aDocument.AddNode(dgsPath, 220, 60, 80, 80, '');
  oStar.PathData := 'M 50 0 L 61 35 L 98 35 L 68 57 L 79 91 L 50 70 L 21 91 L 32 57 L 2 35 L 39 35 Z';
  oStar.Angle := 15;
  // an image node
  aDocument.Images.AddPNG('logo', aLogo);
  oLogo := aDocument.AddNode(dgsImage, 420, 0, 96, 96, '');
  oLogo.ImageKey := 'logo';
end;
```

Consulte también

- Puertos y anclaje
- Temas
- Conectores y enrutamiento
- TsgcDiagramNode

Conectores y enrutamiento

Un TsgcDiagramConnector une un origen y un destino. Cada extremo está anclado a un nodo, mediante SourceNodeId y TargetNodeId, o libre en SourcePoint y TargetPoint. Document.AddConnector(aSource, aTarget) crea un conector del tipo DefaultConnectorKind (dgcOrthogonal de forma predeterminada) y lo enruta al final del conjunto de cambios.

Tipo	Ruta
dgcStraight	Una línea recta, o una polilínea por los Waypoints.
dgcOrthogonal	Segmentos horizontales y verticales. En la edición Registered, el enrutador evita los nodos con un margen y separa los segmentos paralelos; en Community, la ruta es un codo simple.
dgcCurved	Una curva suave, por los Waypoints cuando los hay.

Flechas, línea y etiquetas

- StartArrow (dgaNone de forma predeterminada) y EndArrow (dgaTriangle de forma predeterminada) admiten dgaNone, dgaTriangle, dgaOpenTriangle, dgaDiamond, dgaOpenDiamond o dgaCircle. ArrowSize establece su tamaño.
- Style.Stroke, Style.StrokeWidth y Style.Dash (dgdSolid, dgdDash, dgdDot) definen la línea.
- Labels contiene cualquier número de TsgcDiagramLabel. Position va de 0 en el origen a 1 en el destino, Offset aleja la etiqueta de la línea. El usuario arrastra una etiqueta a lo largo de su conector.

Enrutamiento

El documento enruta cada conector cuyos nodos, extremos, tipo o waypoints cambiaron cuando se ejecuta el EndUpdate más externo, mientras Routing.AutoRoute es True. RouteAll enruta de nuevo todos los conectores. La ruta está en Points; RouteFailed es True cuando el enrutador ortogonal no encontró camino alrededor de los nodos y dibujó una alternativa.

Propiedad de enrutamiento	Predeterminado	Significado
Margin	12	La distancia que una ruta ortogonal mantiene respecto a los nodos.
BendPenalty	40	El coste de una curva, de modo que el enrutador prefiere menos curvas a un camino algo más corto.
Spacing	6	La distancia entre segmentos paralelos que comparten un pasillo.
AutoRoute	True	Enruta al final de cada conjunto de cambios. Desactívelo mientras construye un diagrama grande en el código y luego llame a RouteAll.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model;

procedure AddConnectors(aDocument: TsgcDiagramDocument; aA, aB, aC: TsgcDiagramNode);
```

```
var
  oLink: TsgcDiagramConnector;
begin
  aDocument.Routing.Margin := 16;
  aDocument.BeginUpdate;
  try
    oLink := aDocument.AddConnector(aA, aB);
    oLink.Labels.Add.Text := 'Yes';
    oLink := aDocument.AddConnector(aA, aC);
    oLink.Kind := dgcCurved;
    oLink.StartArrow := dgaDiamond;
    oLink.EndArrow := dgaOpenTriangle;
    oLink.Style.Dash := dgdDash;
    with oLink.Labels.Add do
    begin
      Text := 'No';
      Position := 0.25;
    end;
  finally
    aDocument.EndUpdate;
  end;
end;
```

Consulte también

- Puertos y anclaje
- Ediciones
- Formas
- TsgcDiagramConnector

Puertos y anclaje

Un puerto es un punto de un nodo donde se une el extremo de un conector. Un nodo con una colección Ports vacía usa los puertos predeterminados de su forma: `left`, `top`, `right` y `bottom`, los puntos medios de los lados (las esquinas de una decisión). Añada sus propios puertos para dar a un nodo entradas y salidas tipadas.

TsgcDiagramPort	Significado
Id	Nombre del puerto, único en su nodo. Un conector se refiere a él en SourcePortId o TargetPortId.
X, Y	Posición como fracción del cuadro del nodo, de 0 a 1.
Side	dgpAuto, dgpLeft, dgpTop, dgpRight o dgpBottom: la dirección en que una ruta sale del puerto.
Flow	dgfBoth, dgfIn o dgfOut.
Caption	Texto dibujado dentro del nodo junto al puerto. Las leyendas se ocultan por debajo de zoom 0.4.

Anclaje

- Un extremo de conector con un Id de puerto está anclado a ese puerto.
- Un extremo de conector con un Id de puerto vacío está anclado al contorno del nodo: el extremo sigue al nodo y toca el contorno en la línea hacia el centro.
- Un extremo de conector sin nodo es libre en SourcePoint o TargetPoint.
- Node.PortPoint(aPortId) devuelve la posición del mundo de un puerto.

En la vista

ShowPorts decide cuándo se dibujan los puertos: dgvNever, dgvHover (el valor predeterminado, los puertos del nodo bajo el puntero) o dgvAlways. Un arrastre desde un puerto dibuja un nuevo conector y, mientras se dibuja, cada puerto es un destino. En un nodo seleccionado, los tiradores de redimensión cubren los puertos laterales, así que establezca Tool en dgtConnector para dibujar desde él. Antes de anclar un extremo, la vista dispara OnConnecting, donde puede rechazarlo; una vez creado el conector, dispara OnConnected.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

type
  TNodeEditor = class
  public
    procedure DoConnecting(Sender: TObject; aSource: TsgcDiagramNode;
      const aSourcePort: string; aTarget: TsgcDiagramNode;
      const aTargetPort: string; var aAccept: Boolean);
  end;

function AddFilter(aDocument: TsgcDiagramDocument): TsgcDiagramNode;
begin
  Result := aDocument.AddNode(dgsRoundRect, 0, 0, 160, 80, 'Filter');
  with Result.Ports.Add do
  begin
    Id := 'in';
```

```
    X := 0;
    Y := 0.5;
    Side := dgpLeft;
    Flow := dgfIn;
    Caption := 'rows';
  end;
  with Result.Ports.Add do
  begin
    Id := 'out';
    X := 1;
    Y := 0.5;
    Side := dgpRight;
    Flow := dgfOut;
    Caption := 'result';
  end;
end;

procedure TNodeEditor.DoConnecting(Sender: TObject; aSource: TsgcDiagramNode;
  const aSourcePort: string; aTarget: TsgcDiagramNode;
  const aTargetPort: string; var aAccept: Boolean);
begin
  // an output may only connect to an input
  aAccept := (aSourcePort = 'out') and (aTargetPort = 'in');
end;

procedure Connect(aView: TsgcVCLDiagramView; aEditor: TNodeEditor);
var
  oA, oB: TsgcDiagramNode;
begin
  aView.ShowPorts := dgvAlways;
  aView.OnConnecting := aEditor.DoConnecting;
  oA := AddFilter(aView.Diagram);
  oB := AddFilter(aView.Diagram);
  oB.Left := 260;
  aView.Diagram.AddConnector(oA, oB, 'out', 'in');
end;
```

Consulte también

- Conectores y enrutamiento
- Edición y teclado
- TsgcDiagramPort

Edición y teclado

La vista edita el documento mediante su Controller. Cada acción del usuario que cambia el documento es un paso de deshacer, y Esc durante un arrastre lo cancela.

Herramientas

Herramienta	Efecto
dgtSelect	El valor predeterminado. Haga clic para seleccionar, arrastre para mover, los tiradores redimensionan y giran, un arrastre en un espacio vacío dibuja un rectángulo de selección, un arrastre desde un puerto dibuja un conector.
dgtPan	Un arrastre con el botón izquierdo desplaza la vista.
dgtConnector	Un arrastre desde cualquier nodo dibuja un conector.
dgtShape	Un clic o un arrastre crea un nodo de la forma de ShapeTool. La herramienta sigue activa después de cada nodo; Esc vuelve a dgtSelect. Un clic en un elemento de la paleta establece esta herramienta.

Ratón

- Shift y un clic añaden a la selección, Ctrl y un clic alternan un elemento, Alt y un clic seleccionan un miembro de un grupo.
- Shift al redimensionar mantiene la proporción; Shift al girar ajusta el ángulo en pasos de 15 grados.
- La rueda desplaza en vertical, Shift y la rueda en horizontal, Ctrl y la rueda hacen zoom alrededor del puntero.
- El botón central, o el botón izquierdo con la barra espaciadora pulsada, desplaza la vista.
- El botón derecho selecciona el elemento bajo el puntero, listo para un menú emergente.
- Un doble clic en un nodo edita su texto y dispara OnItemDblClick.

Teclado

Teclas	Acción
Ctrl+Plus, Ctrl+Minus	Zoom de acercamiento y alejamiento alrededor del centro de la vista.
Ctrl+0, Ctrl+1	Zoom 1, zoom para ajustar.
Ctrl+A, Ctrl+C	Seleccionar todo, copiar.
Ctrl+X, Ctrl+V, Ctrl+D	Cortar, pegar, duplicar.
Ctrl+Z, Ctrl+Y o Ctrl+Shift+Z	Deshacer, rehacer.
Ctrl+G, Ctrl+Shift+G	Agrupar, desagrupar.
Delete	Elimina la selección (los elementos bloqueados permanecen).
Teclas de flecha	Mueve la selección según GridSize; con Alt, una unidad del mundo.

F2, o cualquier carácter	Edita el texto del nodo seleccionado.
Tab, Shift+Tab	Selecciona el elemento siguiente o el anterior.
Esc	Cancela un arrastre, sale de la herramienta actual o borra la selección.

En el modo ReadOnly, solo funcionan las teclas de zoom, Ctrl+A, Ctrl+C, Tab y Esc.

Edición de texto

El texto se edita en un memo sobre el nodo. Enter o un clic en otro lugar confirma, Esc cancela, Shift+Enter inicia una nueva línea. OnTextEditing puede rechazar la edición estableciendo aAllow en False, y OnTextEdited se dispara tras una confirmación. EditText(aNode) inicia el editor desde el código.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

type
  TEditorEvents = class
  public
    procedure DoTextEditing(Sender: TObject; aNode: TsgcDiagramNode;
      var aAllow: Boolean);
  end;

procedure TEditorEvents.DoTextEditing(Sender: TObject; aNode: TsgcDiagramNode;
  var aAllow: Boolean);
begin
  // the terminators keep their text
  aAllow := aNode.Shape <> dgsTerminator;
end;

procedure DrawDecisions(aView: TsgcVCLDiagramView; aEvents: TEditorEvents);
begin
  aView.OnTextEditing := aEvents.DoTextEditing;
  aView.ShapeTool := dgsDecision;
  aView.Tool := dgtShape;
end;
```

Consulte también

- Ajuste y guías
- Deshacer y portapapeles
- Modo visor
- TsgcDiagramController
- Eventos

Ajuste y guías

Dos ayudas alinean los nodos mientras el usuario los mueve, redimensiona o crea.

Propiedad	Efecto
Diagram.GridSize	Espaciado de la cuadrícula en unidades del mundo, 10 de forma predeterminada. Las teclas de flecha mueven la selección esa cantidad.
ShowGrid	Dibuja la cuadrícula en la vista.
SnapToGrid	Ajusta a la cuadrícula los nodos movidos, redimensionados y creados, y las formas soltadas.
SnapToGuides	Muestra guías de alineación: mientras un nodo se mueve o se redimensiona, sus bordes y líneas centrales se ajustan a los bordes y líneas centrales de los nodos cercanos y se dibuja una línea guía.

- Una guía captura un borde o un centro a menos de 5 píxeles del dispositivo, respecto a los nodos a menos de 400 píxeles del dispositivo.
- Las guías se aplican a los nodos que no están girados.
- Shift al redimensionar desactiva las guías en ese arrastre.
- Alt con una tecla de flecha mueve la selección una unidad del mundo, fuera de la cuadrícula.

Ejemplo

```
uses
  sgcDiagram_VCL;

procedure SetupSnapping(aView: TsgcVCLDiagramView);
begin
  aView.Diagram.GridSize := 20;
  aView.ShowGrid := True;
  aView.SnapToGrid := True;
  aView.SnapToGuides := True;
end;
```

Consulte también

- Edición y teclado
- Formas

Deshacer y portapapeles

El documento mantiene una pila de deshacer y una pila de rehacer. Cada acción del usuario en la vista es un paso. En el código, BeginTransaction(aCaption) y EndTransaction agrupan cualquier número de cambios en un paso, y Rollback descarta los cambios de la transacción abierta.

- Undo, Redo, CanUndo, CanRedo, UndoCaption y RedoCaption actúan sobre las pilas; la vista y el controlador también tienen Undo y Redo.
- UndoLimit (100 de forma predeterminada) es el número de pasos que se conservan.
- Un cambio hecho por código fuera de una transacción borra las dos pilas, de modo que un deshacer nunca restaura un estado obsoleto.
- BeginUndoSuspend y EndUndoSuspend envuelven una reconstrucción masiva que no debe registrarse: borran las pilas y no generan paso, y las vistas siguen recibiendo notificaciones. Se pueden anidar.
- Document.Clear no es un paso de deshacer.

Portapapeles

El portapapeles es interno a la aplicación. CopySelection devuelve los nodos seleccionados, los hijos de los contenedores seleccionados, los conectores cuyos dos extremos se copian y las imágenes que usan, como un fragmento JSON, y lo guarda en el portapapeles interno. Paste(aJSON) inserta un fragmento como un único paso de deshacer y selecciona los elementos pegados; Paste sin argumento pega el portapapeles interno, 20 unidades del mundo más allá por cada pegado repetido. La vista tiene CopyToClipboard, CutToClipboard, PasteFromClipboard y DeleteSelection, y Ctrl+D duplica la selección.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model;

procedure AddApproval(aDocument: TsgcDiagramDocument);
begin
  aDocument.BeginTransaction('Add approval');
  try
    aDocument.AddNode(dgsDecision, 0, 0, 160, 90, 'Approved?');
    aDocument.AddNode(dgsProcess, 0, 140, 160, 60, 'Notify');
    aDocument.EndTransaction;
  except
    aDocument.Rollback;
    raise;
  end;
  // one step: undo removes both nodes
  if aDocument.CanUndo then
    aDocument.Undo;
end;

procedure DuplicateSelection(aDocument: TsgcDiagramDocument);
var
  vFragment: string;
begin
  vFragment := aDocument.CopySelection;
  aDocument.Paste(vFragment);
end;
```

Consulte también

- Serialización
- Edición y teclado

Capas

Las capas organizan los elementos de un diagrama, por ejemplo una capa de fondo con marcos y notas bajo la capa de trabajo. Cada nodo y conector indica su capa en la propiedad Layer; un nombre vacío es la capa predeterminada.

TsgcDiagramLayer	Significado
Name	Nombre que usa la propiedad Layer de los elementos.
Visible	False oculta los elementos de la capa en la vista y en la exportación.
Locked	True impide que los elementos de la capa se seleccionen y se editen.

- Layers.Add, Layers.Find(aName) y los Items de la colección gestionan las capas. Añadir, eliminar y renombrar una capa son pasos de deshacer; Visible y Locked no lo son.
- Document.LayerVisible(aName) y LayerLocked(aName) indican cómo se trata una capa.
- Un único elemento también se puede bloquear con su propia propiedad Locked.

Edición Community

Sin SGC_DIAGRAM_ADVANCED solo se puede añadir una capa: un segundo Layers.Add genera EsgcDiagramError. Un archivo con más capas aún se carga y se guarda sin cambios, pero cada capa se dibuja, se exporta y se edita como visible y desbloqueada.

Ejemplo

```
uses
  sgcDiagram_Types, sgcDiagram_Model;

procedure AddBackground(aDocument: TsgcDiagramDocument);
var
  oLayer: TsgcDiagramLayer;
  oNote: TsgcDiagramNode;
begin
  oLayer := aDocument.Layers.Add;
  oLayer.Name := 'Background';
  oLayer.Locked := True;
  oNote := aDocument.AddNode(dgsText, 0, -60, 300, 40, 'Draft, not approved');
  oNote.Layer := 'Background';
end;
```

Consulte también

- Ediciones
- Modo visor
- TsgcDiagramLayer

Serialización

Un documento se guarda y se carga como JSON. El formato es el mismo en todas las plataformas y en todas las ediciones.

Método	Efecto
SaveToJSON, LoadFromJSON	Hacia y desde una cadena.
SaveToFile, LoadFromFile	Hacia y desde un archivo.
SaveToStream, LoadFromStream	Hacia y desde un stream.
CopySelection, Paste	Un fragmento del documento, consulte Deshacer y portapapeles.

- La salida es canónica: un orden de claves fijo, propiedades con su valor predeterminado omitidas, sin sangría. El mismo documento siempre genera el mismo texto.
- Las imágenes del documento se almacenan como datos PNG en el archivo.
- El lector es tolerante: las claves desconocidas y los nombres de enumeración desconocidos se ignoran. Un archivo escrito por una versión de formato más nueva genera EsgcDiagramError, y un error de análisis indica su posición.
- Cargar reemplaza el contenido del documento. Las rutas de los conectores almacenadas en el archivo se conservan.
- Una vista almacena su diagrama en el archivo dfm o fmx como la propiedad DiagramJSON, solo cuando el diagrama no está vacío.

Ejemplo

```
uses
  SysUtils,
  sgcDiagram_Model;

procedure SaveAndReload(aDocument: TsgcDiagramDocument; const aFileName: string);
var
  vJSON: string;
begin
  aDocument.SaveToFile(aFileName);
  aDocument.LoadFromFile(aFileName);
  // or through a string, for a database field
  vJSON := aDocument.SaveToJSON;
  aDocument.LoadFromJSON(vJSON);
end;
```

Un archivo pequeño

Dos nodos y un conector, con sangría aquí para facilitar la lectura:

```
{
  "format": "sgcDiagram",
  "version": 1,
  "nodes": [
    {
      "id": "n1",
      "shape": "terminator",
      "x": 40,
      "y": 40,
      "w": 140,
      "h": 50,
```

```
      "text": "Start"
    },
    {
      "id": "n2",
      "shape": "process",
      "x": 40,
      "y": 140,
      "w": 140,
      "text": "Save"
    }
  ],
  "connectors": [
    {
      "id": "c3",
      "source": {
        "node": "n1"
      },
      "target": {
        "node": "n2"
      },
      "labels": [
        {
          "text": "Go"
        }
      ],
      "points": [[110, 90], [110, 140]]
    }
  ]
}
```

Consulte también

- Deshacer y portapapeles
- Exportación
- Formato JSON

Exportación

Un diagrama se exporta a PNG y SVG. La imagen cubre los límites del contenido del documento más un margen.

Método	Notas
View.SaveToPNG(aFileName or aStream, aScale)	Exporta el diagrama de la vista con su FontFile y OnDrawNode aplicados. aScale es el número de píxeles por unidad del mundo, 1 de forma predeterminada.
View.SaveToSVG(aFileName or aStream, aScale)	Lo mismo como SVG.
TsgcDiagramRenderer.SaveToPNGFile, SaveToPNGStream	(aDocument, aFileName o aStream, aScale = 1, aMargin = 20). No necesita vista.
TsgcDiagramRenderer.SaveToSVGFile, SaveToSVGStream	Lo mismo como SVG.
TsgcDiagramRenderer.RenderTo	(aCanvas, aDocument, aViewport) pinta una parte del documento en un TsgcUICanvas de sgcUICore.

- La exportación usa el tema del documento, de modo que un diagrama oscuro se exporta oscuro.
- El renderizador funciona en una aplicación de consola, en un servicio y en Linux64; consulte Inicio rápido sin ventana.
- La exportación a PDF y la impresión no forman parte de la versión 1.

Ejemplo

```
uses
  SysUtils,
  sgcDiagram_VCL;

procedure ExportView(aView: TsgcVCLDiagramView; const aFolder: string);
begin
  // 2 pixels per world unit, for a sharp image on a slide
  aView.SaveToPNG(IncludeTrailingPathDelimiter(aFolder) + 'diagram.png', 2);
  aView.SaveToSVG(IncludeTrailingPathDelimiter(aFolder) + 'diagram.svg');
end;
```

Consulte también

- Temas
- Serialización
- TsgcDiagramRenderer

Temas

Las vistas, las paletas y la exportación se dibujan con los temas de sgcUICore, de modo que un diagrama coincide con los demás productos sgcUI en el mismo formulario.

- **Theme** de la vista, la paleta y el documento es utLight (predeterminado), utDark o utCustom. El Theme de una vista establece el Theme de su Diagram.
- **Style** de cada nodo y conector: Fill, Stroke, StrokeWidth, Dash, CornerRadius y Text (Size, Bold y Color). Un color es un TsgcUIColor ARGB, por ejemplo sgcUIARGB($FF, $00, $96, $88); 0 conserva el color del tema, que depende de la forma.
- **NodeStyle** y **ConnectorStyle** del documento son las plantillas que se copian en cada nuevo nodo y conector.
- **FontFile** de la vista dibuja el texto con un archivo TrueType en lugar de la fuente incrustada en sgcUICore.
- **OnDrawNode** de la vista le permite pintar un nodo usted mismo, o añadir al dibujo predeterminado: establezca aDefault en False para omitirlo.

Ejemplo

```
uses
  sgcUICore_Theme, sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

procedure ApplyCorporateStyle(aView: TsgcVCLDiagramView);
var
  oNode: TsgcDiagramNode;
begin
  aView.Theme := utDark;
  // new nodes get a teal fill and bold white text
  aView.Diagram.NodeStyle.Fill := sgcUIARGB($FF, $00, $96, $88);
  aView.Diagram.NodeStyle.Text.Bold := True;
  aView.Diagram.NodeStyle.Text.Color := sgcUIARGB($FF, $FF, $FF, $FF);
  aView.Diagram.ConnectorStyle.StrokeWidth := 2;
  // one node with its own style
  oNode := aView.Diagram.AddNode(dgsRoundRect, 0, 0, 160, 60, 'Alert');
  oNode.Style.Fill := sgcUIARGB($FF, $E5, $39, $35);
  oNode.Style.Dash := dgdDash;
  // 0 hands the colour back to the theme
  oNode.Style.Stroke := 0;
end;
```

Consulte también

- Exportación
- Formas
- TsgcDiagramStyle

Modo visor

Establezca ReadOnly en True para mostrar un diagrama que el usuario puede explorar pero no cambiar, como un mapa de procesos en un panel o un organigrama generado.

- El usuario puede seleccionar elementos, hacer zoom con Ctrl y la rueda o las teclas, desplazarse con la rueda, el botón central, la barra espaciadora o la herramienta de desplazamiento, y copiar la selección.
- No se mueve, redimensiona, conecta, elimina, pega ni edita nada, y la paleta no puede soltar formas.
- El puntero se convierte en una mano sobre un elemento, y OnItemClick, OnItemDblClick y OnHover informan del elemento, de modo que la aplicación puede abrir una página de detalles o mostrar una sugerencia.
- ZoomToFit, ZoomToSelection y CenterOn mueven la vista desde el código; MinZoom (0.05) y MaxZoom (8) acotan el zoom.

Para proteger solo algunos elementos, deje ReadOnly en False y establezca Locked en esos elementos o en su capa.

Ejemplo

```
uses
  SysUtils,
  sgcDiagram_Types, sgcDiagram_Model, sgcDiagram_VCL;

type
  TViewerEvents = class
  public
    procedure DoItemClick(Sender: TObject; aItem: TsgcDiagramItem);
  end;

procedure TViewerEvents.DoItemClick(Sender: TObject; aItem: TsgcDiagramItem);
begin
  if aItem is TsgcDiagramNode then
    // Data holds an application key, for example a record id
    Writeln('open ' + TsgcDiagramNode(aItem).Data);
end;

procedure ShowProcess(aView: TsgcVCLDiagramView; aEvents: TViewerEvents;
  const aFileName: string);
begin
  aView.Diagram.LoadFromFile(aFileName);
  aView.ReadOnly := True;
  aView.OnItemClick := aEvents.DoItemClick;
  aView.ZoomToFit;
end;
```

Consulte también

- Edición y teclado
- Capas
- TsgcVCLDiagramView
- TsgcFMXDiagramView

Descripción general de .NET

sgcDiagram .NET es un port administrado en C# de la biblioteca sgcDiagram de Delphi, no un contenedor alrededor de ella. Las clases de C# viven en el espacio de nombres **esegece.sgcDiagram** y conservan los nombres de Delphi, de modo que **TsgcDiagramDocument**, **TsgcDiagramNode**, **TsgcDiagramConnector**, **TsgcDiagramController**, **TsgcDiagramRenderer** y las enumeraciones se leen igual en ambos lenguajes. La vista y la paleta siguen el patrón de nombres de Delphi en cada host: **TsgcVCLDiagramView** es **TsgcWinFormsDiagramView**, **TsgcWpfDiagramView** y **TsgcAvaloniaDiagramView**, y **TsgcVCLDiagramPalette** es **TsgcWinFormsDiagramPalette**, **TsgcWpfDiagramPalette** y **TsgcAvaloniaDiagramPalette**. Los controles se dibujan con el port .NET de **sgcUICore**, el mismo lienzo, las mismas fuentes y los mismos temas claro y oscuro que los demás productos de la suite visual de eSeGeCe.
El documento, la pila de deshacer, el formato JSON, el enrutador, los diseños, el pintor y el controlador son el código de Delphi portado línea a línea. Un diagrama guardado por una aplicación .NET se abre en una aplicación Delphi y al revés, el texto JSON y la exportación SVG son los mismos que en Delphi, y los diseños y las rutas dan las mismas coordenadas.

Paquetes

La versión se divide en cinco paquetes NuGet, de modo que un proyecto solo incorpora el host que usa.

PackageId	Contiene	Frameworks de destino
esegece.sgcDiagram	El documento con sus nodos, conectores, puertos, etiquetas, capas, imágenes y estilos, deshacer y rehacer, el formato JSON y el portapapeles interno, el enrutador, los diseños por capas y en árbol, el modelo de la paleta, el controlador, el pintor, la exportación SVG de **TsgcDiagramRenderer** y **TsgcDiagramDataBinding**. Sin dependencia nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDiagram.Rendering	La salida PNG: los métodos de extensión **SaveToPNGStream** y **SaveToPNGFile** de **TsgcDiagramRenderer** y el raster de la caché de la vista, dibujados en el lienzo SkiaSharp de esegece.sgcUICore.Rendering, en Windows, Linux y macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDiagram.WinForms	**TsgcWinFormsDiagramView** y **TsgcWinFormsDiagramPalette**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDiagram.Wpf	**TsgcWpfDiagramView** y **TsgcWpfDiagramPalette**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDiagram.Avalonia	**TsgcAvaloniaDiagramView** y **TsgcAvaloniaDiagramPalette**, en Windows, Linux y macOS con Avalonia 11.	net8.0, net9.0

Los tres paquetes de host dependen de esegece.sgcDiagram.Rendering, que depende de esegece.sgcDiagram, así que una línea `PackageReference` trae todo lo que necesita un host. Véase Instalación desde NuGet.

Ediciones

Edición	Ids de paquete	Incluye
Registered	esegece.sgcDiagram, .Rendering, .WinForms, .Wpf y .Avalonia	Todo: formas, conectores, puertos y anclaje, edición, ajuste y guías, deshacer y portapapeles, capas, JSON, exportación PNG y SVG, temas, el modo visor y las funciones avanzadas: los diseños por capas y en árbol, el enrutamiento ortogonal con evitación de obstáculos y ajuste, el enlace de datos y más de una capa.
Community	Los mismos ids con el sufijo .Community, por ejemplo esegece.sgcDiagram.WinForms.Community.	Todo excepto las funciones avanzadas. **TsgcDiagramLayeredLayout**, **TsgcDiagramTreeLayout** y **TsgcDiagramDataBinding** no están en los paquetes Community, los conectores **dgcOrthogonal** se enrutan como codos simples, y **Layers.Add** produce **EsgcDiagramError** cuando ya existe una capa.

Es la misma división que las ediciones de Delphi, y un archivo de diagrama es el mismo en ambas, véase Ediciones.

Hosts

Host	Equivalente en Delphi	Plataformas
WinForms	VCL (TsgcVCLDiagramView, TsgcVCLDiagramPalette)	Windows, .NET Framework 4.7.2 y .NET 8 y 9.
WPF	VCL y FMX	Windows, .NET Framework 4.7.2 y .NET 8 y 9. Los controles se pueden declarar en XAML.
Avalonia	FMX (TsgcFMXDiagramView, TsgcFMXDiagramPalette), con zoom por pellizco y desplazamiento con un dedo en una pantalla táctil	Windows, Linux y macOS, .NET 8 y 9.
Sin ventana	TsgcDiagramDocument and TsgcDiagramRenderer	Cualquier plataforma de los frameworks de destino: JSON y SVG con esegece.sgcDiagram, PNG con esegece.sgcDiagram.Rendering.

Cada host tiene la misma API: las propiedades, métodos y eventos de la vista y la paleta de Delphi con los mismos nombres. Véase Clases de .NET para los nombres de las clases y Diferencias de .NET respecto a Delphi para lo que cambia respecto a Delphi.

Véase también

- Instalación desde NuGet
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Enlace de datos en .NET

- Renderizado sin ventana en .NET
- Demos de .NET
- Plataformas

Instalación desde NuGet

Los paquetes están en nuget.org. Añada el paquete del host, los paquetes del núcleo y del renderizado llegan con él como dependencias.

Edición Registered

```
dotnet add package esegece.sgcDiagram.WinForms
dotnet add package esegece.sgcDiagram.Wpf
dotnet add package esegece.sgcDiagram.Avalonia
```

Edición Community

```
dotnet add package esegece.sgcDiagram.WinForms.Community
dotnet add package esegece.sgcDiagram.Wpf.Community
dotnet add package esegece.sgcDiagram.Avalonia.Community
```

O añada la referencia al archivo de proyecto a mano:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcDiagram.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie en un proyecto los paquetes Registered o los Community, nunca los dos: contienen los mismos tipos en el mismo espacio de nombres.

Configuración del proyecto

- WinForms: un proyecto `net8.0-windows` con `<UseWindowsForms>true</UseWindowsForms>`, o un proyecto .NET Framework 4.7.2.
- WPF: un proyecto `net8.0-windows` con `<UseWPF>true</UseWPF>`, o un proyecto .NET Framework 4.7.2. Asigne el espacio de nombres en XAML con `xmlns:sgc="clr-namespace:esegece.sgcDiagram;assembly=esegece.sgcDiagram.Wpf"`.
- Avalonia: un proyecto `net8.0` o `net9.0` con los paquetes **Avalonia.Desktop** y **Avalonia.Themes.Fluent** de Avalonia 11.3, como cualquier aplicación de escritorio Avalonia. El editor de texto en el lugar es un **TextBox** de Avalonia y toma su aspecto del tema Fluent.
- Sin ventana: esegece.sgcDiagram solo basta para el documento, el JSON y el SVG. Añada esegece.sgcDiagram.Rendering para PNG, que trae SkiaSharp y sus recursos nativos para Windows y Linux.

En C# todo host necesita `using esegece.sgcDiagram;`, y `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUINotifyEvent** y los demás tipos de sgcUICore.

Véase también

- Descripción general de .NET
- Inicio rápido WinForms
- Inicio rápido WPF
- Inicio rápido Avalonia
- Demos de .NET

Inicio rápido WinForms

Este inicio rápido construye un formulario en código con una paleta de formas a la izquierda, una vista de diagrama ocupando el resto y un pequeño diagrama de flujo en la vista. Arrastre una forma desde la paleta a la vista, o haga clic en ella y luego en la vista, y conecte dos nodos desde un puerto.

Pasos

1. Cree un proyecto Windows Forms para `net8.0-windows` y añada el paquete **esegece.sgcDiagram.WinForms** (**esegece.sgcDiagram.WinForms.Community** en la edición Community).
2. Añada `using esegece.sgcDiagram;` y `using esegece.sgcUICore;`.
3. Cree un **TsgcWinFormsDiagramView** y una **TsgcWinFormsDiagramPalette** y añádalos al formulario. Son controles WinForms normales: Dock, Anchor, TabIndex y Enabled funcionan como de costumbre.
4. Establezca **View** de la paleta en la vista. La paleta carga las formas Flowchart y Basic cuando se crea.
5. Rellene **Diagram**, el **TsgcDiagramDocument** de la vista, dentro de una transacción para que toda la configuración sea un paso de deshacer, y asigne los eventos.

Ejemplo

Notas

- **Tool** elige lo que hace una pulsación (**dgtSelect**, **dgtPan**, **dgtConnector**, **dgtShape** con **ShapeTool**), y **ReadOnly** convierte la vista en un visor, como en Delphi.
- El teclado es el de Delphi, véase Edición y teclado. El editor en el lugar es un **TextBox** de varias líneas, expuesto como **TextEditor**: Enter confirma, Shift+Enter es una nueva línea, Esc cancela.
- **Theme** es por control, así que establézcalo en la vista y en la paleta. La vista lo pasa a su documento.
- El diagrama se almacena en la propiedad oculta **DiagramJSON**, de modo que una vista colocada con el diseñador conserva el diagrama dibujado en ella.
- **SaveToPNG** y **SaveToSVG** escriben el diagrama completo en un archivo o una secuencia, a una escala.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido VCL

Inicio rápido WPF

Los controles WPF se pueden declarar en XAML. Esta ventana tiene la paleta a la izquierda y la vista ocupando el resto; el code behind enlaza la paleta con la vista y añade un pequeño diagrama de flujo.

Pasos

1. Cree un proyecto WPF para `net8.0-windows` y añada el paquete **esegece.sgcDiagram.Wpf** (**esegece.sgcDiagram.Wpf.Community** en la edición Community).
2. Asigne el espacio de nombres en la ventana: `xmlns:sgc="clr-namespace:esegece.sgcDiagram;assembly=esegece.sgcDiagram.Wpf"`.
3. Declare una **TsgcWpfDiagramPalette** y una **TsgcWpfDiagramView** y establezca sus propiedades como atributos.
4. En el code behind establezca **View** de la paleta, rellene **Diagram** y asigne los eventos.

XAML

```
<Window x:Class="QuickStart.QuickStartWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:sgc="clr-namespace:esegece.sgcDiagram;assembly=esegece.sgcDiagram.Wpf"
        Title="sgcDiagram" Width="900" Height="600">
  <DockPanel>
    <sgc:TsgcWpfDiagramPalette x:Name="palShapes" DockPanel.Dock="Left" Width="180" />
    <sgc:TsgcWpfDiagramView x:Name="dgmFlow" ShowGrid="True" SnapToGrid="True" ShowPorts="dgvHover" />
  </DockPanel>
</Window>
```

Code behind

Notas

- Las propiedades son propiedades CLR simples, por lo que XAML las establece como valores de atributo. **View** de la paleta toma el objeto vista, establézcalo en código.
- Un diagrama se puede dar en XAML con el atributo **DiagramJSON**. El texto JSON empieza con una llave, así que escríbalo después del escape `{}`.
- Las coordenadas de la API, incluidas **ScreenToDiagram** y **DiagramToScreen**, son píxeles de dispositivo del control, es decir, píxeles independientes del dispositivo por la escala de DPI.
- Llame a **Dispose** de la vista cuando se cierre la ventana: detiene el temporizador de la vista y libera su controlador.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido VCL

Inicio rápido Avalonia

Los controles Avalonia se ejecutan en Windows, Linux y macOS. Este inicio rápido construye una ventana en código con la paleta a la izquierda y la vista ocupando el resto, en el tema oscuro.

Pasos

1. Cree un proyecto de escritorio Avalonia para `net8.0` con Avalonia 11.3 y el tema Fluent, y añada el paquete **esegece.sgcDiagram.Avalonia** (**esegece.sgcDiagram.Avalonia.Community** en la edición Community).
2. Cree un **TsgcAvaloniaDiagramView** y una **TsgcAvaloniaDiagramPalette**, establezca **View** de la paleta y coloque ambos en un panel.
3. Rellene **Diagram** y asigne los eventos.

Ejemplo

Notas

- Los eventos de los controles Avalonia son propiedades delegado, que se asignan con `=`, mientras que los controles WinForms y WPF tienen eventos de C#, que se asignan con `+=`.
- **Theme** de los controles Avalonia oculta la propiedad **Theme** de Avalonia (el tema del control) y recibe un **TsgcUITheme**.
- En una pantalla táctil un pellizco hace zoom y un dedo sobre un espacio vacío desplaza; un dedo sobre un nodo lo arrastra, como en la vista FMX.
- Un arrastre de la paleta usa `DragDrop.DoDragDropAsync` con el formato de aplicación **DragFormat** de la vista (**CS_DIAGRAM_AVALONIA_DRAG_FORMAT**). **DragPayloadOf** y **CreateDragData** leen y construyen esos datos para un origen de arrastre propio.
- Avalonia no libera los controles, así que llame a **Dispose** de la vista cuando se cierre la ventana.

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Clases de .NET
- Diferencias de .NET respecto a Delphi
- Inicio rápido FMX

Clases de .NET

Las clases de C# tienen los nombres de las clases de Delphi. Las clases del modelo, los motores y el pintor son las mismas en cada host; los controles tienen una clase por host.

Controles

Delphi	WinForms	WPF	Avalonia
TsgcVCLDiagramView, TsgcFMXDiagramView	TsgcWinFormsDiagramView	TsgcWpfDiagramView	TsgcAvaloniaDiagramView
TsgcVCLDiagramPalette, TsgcFMXDiagramPalette	TsgcWinFormsDiagramPalette	TsgcWpfDiagramPalette	TsgcAvaloniaDiagramPalette
Clase base	TsgcUIWinFormsControl	TsgcUIWpfControl	TsgcUIAvaloniaControl

La vista tiene las propiedades de Delphi **Diagram**, **Controller**, **Tool**, **ShapeTool**, **ReadOnly**, **Zoom**, **MinZoom**, **MaxZoom**, **ShowGrid**, **SnapToGrid**, **SnapToGuides**, **ShowPorts**, **Theme** y **FontFile**, además de **DiagramJSON** y **TextEditor**; los métodos **ZoomToFit**, **ZoomToSelection**, **CenterOn**, **ScreenToDiagram**, **DiagramToScreen**, **Undo**, **Redo**, **CopyToClipboard**, **CutToClipboard**, **PasteFromClipboard**, **DeleteSelection**, **EditText**, **SaveToPNG** y **SaveToSVG**; y los eventos **OnChange**, **OnSelectionChange**, **OnItemClick**, **OnItemDblClick**, **OnHover**, **OnConnecting**, **OnConnected**, **OnDrawNode**, **OnTextEditing**, **OnTextEdited** y **OnZoomChange**. La paleta tiene **Model**, **View**, **Items**, **IconSize**, **ShowCaptions**, **Theme**, **DragPayload**, **LoadDefaultItems** y **OnItemClick**.

Modelo y motores

En el paquete esegece.sgcDiagram:

Clase	Descripción
TsgcDiagramDocument	El diagrama: **Nodes**, **Connectors**, **Layers**, **Images**, **Selection**, **NodeStyle**, **ConnectorStyle**, **Routing**, **Theme**, **GridSize**; **AddNode**, **AddConnector**, **BeginTransaction**, **EndTransaction**, **Undo**, **Redo**, **RouteAll**, **SaveToJSON**, **LoadFromJSON**, **SaveToFile**, **LoadFromFile**, **CopySelection**, **Paste**. Una vista crea el suyo, un programa sin ventana crea uno con `new TsgcDiagramDocument()`.
TsgcDiagramNode, TsgcDiagramConnector	Los elementos, con **Style**, **Layer** y **Data**; un nodo tiene **Ports**, un conector tiene **Labels**, como en Delphi.
TsgcDiagramPort, TsgcDiagramLabel, TsgcDiagramLayer, TsgcDiagramImage, TsgcDiagramStyle	Las partes de los elementos y del documento. Las colecciones son **TsgcDiagramNodes**, **TsgcDiagramConnectors**, **TsgcDiagramPorts**, **TsgcDiagramLabels** y **TsgcDiagramLayers**, con `Add()`, `Count` y un indexador.
TsgcDiagramUndoStack	La pila de deshacer de un documento, a la que se accede mediante **UndoStack**.
TsgcDiagramController	La lógica de edición compartida por los hosts: prueba de impacto, los estados de arrastre, el mapa de teclas, el ajuste, las guías y los drops. Una vista la expone como **Controller**.

TsgcDiagramPaletteModel, TsgcDiagramPaletteItem	La paleta sin host: elementos, categorías, diseño, pintado, prueba de impacto y la carga útil de arrastre.
TsgcDiagramPainter, TsgcDiagramViewCache	El pintor del documento y el raster en caché de una vista.
TsgcDiagramRenderer	Exportación sin ventana. **SaveToSVGStream** y **SaveToSVGFile** están en esegece.sgcDiagram, **SaveToPNGStream** y **SaveToPNGFile** son métodos de extensión en esegece.sgcDiagram.Rendering.
TsgcDiagramLayeredLayout, TsgcDiagramTreeLayout	Los diseños automáticos. Edición Registered.
TsgcDiagramDataBinding	Construye un diagrama a partir de un `DataTable`, véase Enlace de datos en .NET. Edición Registered.
EsgcDiagramError	La excepción de la biblioteca.

Clases estáticas

Las constantes y funciones a nivel de unidad de Delphi son miembros estáticos de una clase con el nombre de la unidad: **sgcDiagram_JSON** (**sgcDiagramSaveJSON**, **sgcDiagramLoadJSON**, **sgcDiagramCopy**, **sgcDiagramPaste**), **sgcDiagram_Render**, **sgcDiagram_Controller** (las constantes **CS_DIAGRAM_*** y los códigos de tecla **CS_DIAGRAM_KEY_***), **sgcDiagram_Palette** y **sgcDiagram_Types**. **sgcDiagram_Render_Rendering.RegisterRaster** instala el raster de Skia; los controles lo llaman en su constructor. El formato de arrastre de la paleta es **sgcDiagram_WinForms.CS_DIAGRAM_DRAG_FORMAT**, **sgcDiagram_Wpf.CS_DIAGRAM_DRAG_FORMAT** y **TsgcAvaloniaDiagramView.CS_DIAGRAM_AVALONIA_DRAG_FORMAT**, todos `esegece.sgcDiagram.Shape`.

Enumeraciones

Las enumeraciones y sus valores conservan los nombres de Delphi: **TsgcDiagramShape** (**dgsRectangle**, **dgsDecision** y las demás formas), **TsgcDiagramConnectorKind**, **TsgcDiagramArrow**, **TsgcDiagramTool**, **TsgcDiagramShowPorts**, **TsgcDiagramDirection**, **TsgcDiagramTreeStyle**, **TsgcDiagramBindingLayout**. Un valor se escribe con su tipo, por ejemplo `TsgcDiagramShape.dgsProcess`. Los conjuntos de Delphi son enumeraciones `[Flags]`: **TsgcDiagramChanges** y **TsgcDiagramShiftState**.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- TsgcDiagramDocument
- TsgcDiagramController
- TsgcVCLDiagramView

Diferencias de .NET respecto a Delphi

La API de C# sigue a la API de Delphi nombre por nombre. Lo que cambia viene del lenguaje y de los frameworks:

Delphi	.NET
Los eventos son punteros a método, Sender es el componente.	Los controles WinForms y WPF tienen eventos de C# (`+=`), Sender es el control. Los controles Avalonia, el documento, el controlador, el modelo de la paleta y el enlace de datos tienen propiedades delegado (`=`).
Parámetros `var` de los eventos (**OnTextEditing**, **OnConnecting**, **OnDrawNode**).	Parámetros `ref` . Una lambda con un parámetro `ref` nombra los tipos de todos sus parámetros.
Propiedades indexadas y valores leídos mediante una función.	Métodos: **ContentBounds()**, **UndoCaption()**, **CanUndo()**, **Viewport()** del controlador, **GetCollapsed**, **SetCollapsed**, **GetCategory**, **MaxScroll()** y **Payload(i)** del modelo de la paleta. Un elemento de colección se añade con `Add()` .
Argumentos **TList**.	`List<TsgcDiagramItem>` , `List<TsgcDiagramNode>` and `List<TsgcDiagramConnector>` .
UndoStack es un **TObject**.	Sigue siendo un `object` : haga el cast a **TsgcDiagramUndoStack**.
Free.	El recolector de basura. Las vistas WPF y Avalonia son `IDisposable` , llame a **Dispose** cuando se cierre la ventana; una vista WinForms se libera con su formulario. **TsgcDiagramController** y **TsgcDiagramDataBinding** son `IDisposable` .
El diagrama transmitido en el archivo DFM o FMX.	La propiedad oculta **DiagramJSON**, escrita por el diseñador de WinForms solo cuando el diagrama no está vacío, y utilizable como atributo XAML.
Códigos de tecla y **TShiftState**.	Códigos de tecla virtuales `int` (**CS_DIAGRAM_KEY_***; los valores **Keys** de WinForms son los mismos) y la enumeración `[Flags]` **TsgcDiagramShiftState**.
Los colores son **TsgcUIColor**.	Valores ARGB `uint` , por ejemplo `0xFFDBEAFE` ; 0 es el color del tema.
Exportación PNG de **TsgcDiagramRenderer**.	Métodos de extensión de esegece.sgcDiagram.Rendering, dibujados con Skia. Los archivos PNG no son idénticos byte a byte a los de Delphi; el texto SVG y JSON sí lo es.
TsgcDiagramDataBinding es un componente con un **TDataSource**.	Una clase simple con un `DataTable` , véase Enlace de datos en .NET.

Ejemplo

Eventos con parámetros `ref`, un conjunto de flags y los métodos que sustituyen a propiedades de Delphi:

Comportamiento que es el mismo

- Unidades del mundo, el eje y hacia abajo, ángulos en grados en sentido horario, la cuadrícula, el ajuste y el orden de impacto.

- Los rótulos de deshacer, un paso de deshacer por transacción, el formato del portapapeles interno y la carga útil de arrastre de la paleta.
- El formato JSON, las rutas, los diseños y la exportación SVG, con los mismos números y texto que en Delphi.

Véase también

- Descripción general de .NET
- Clases de .NET
- Eventos
- Formato JSON

Enlace de datos en .NET

El **TsgcDiagramDataBinding** de Delphi construye un diagrama a partir de un TDataSet: un nodo por registro y un conector de cada registro a su padre. En .NET la misma clase lee un `System.Data.DataTable`, la tabla que rellena un `DataAdapter` de cualquier proveedor de ADO.NET, o una **IsgcUITable** de sgcUICore. Es una característica de la edición Registered.

Propiedad	Descripción
DataSource	El `DataTable` de los registros.
Table	Una **IsgcUITable**, que se lee antes que **DataSource** cuando se asigna.
KeyField, ParentField, ParentSeparator, LabelField	La columna de la clave, la columna de la clave del padre (varios padres separados por **ParentSeparator**, un punto y coma por defecto) y la columna del texto del nodo.
Target, Document	**Target** es una vista de diagrama, o cualquier objeto que implemente **IsgcDiagramDocumentOwner**; **Document** es un documento en el que escribir directamente, para un programa sin vista.
Layout, Direction, TreeStyle	El diseño del diagrama reconstruido (**dgbAuto**, **dgbTree**, **dgbLayered**, **dgbNone**) y sus opciones.
NodeShape, NodeWidth, NodeHeight, ConnectorKind	La forma y el tamaño de los nodos y el tipo de los conectores.
LiveUpdates, Active	El diagrama se reconstruye tras un cambio de la tabla mientras **LiveUpdates** sea true; al establecer **Active** se construye.
OnCreateNode, OnCreateConnector	Se produce por cada nodo y conector creado, con el índice de la fila. **CurrentRow** es el `DataRow` del nodo mientras se ejecuta **OnCreateNode**.

Ejemplo

Diferencias respecto a TDataSet

- Un `DataTable` no tiene cursor, por lo que **OnCreateNode** lee la fila mediante **CurrentRow** en lugar de los campos del dataset.
- El enlace escucha los eventos de la tabla: una fila añadida, cambiada o eliminada, un cambio rechazado y una tabla vaciada reconstruyen el diagrama una vez. `AcceptChanges` y moverse por las filas no reconstruyen nada. Una fila en edición (`BeginEdit`) se usa cuando se ejecuta su `EndEdit`. Las filas eliminadas y desasociadas se omiten.
- La reconstrucción se encola, como el **ForceQueue** de Delphi. En una aplicación WinForms, WPF o Avalonia se encola en el **SynchronizationContext** actual cuando se creó el enlace, así que no hay que llamar a nada. Un programa de consola o un servicio no tiene contexto: llame a **CheckQueued** en el hilo del documento, el **CheckSynchronize** de Delphi, o a **Refresh** para reconstruir de inmediato.
- El enlace no es un componente: es `IDisposable`, y una tabla o destino liberado se desenlaza solo.

Enlazar los controles

Las propiedades de los controles son propiedades CLR simples, no propiedades de dependencia ni con estilo, porque un diagrama es un modelo completo y no un único valor. En una aplicación MVVM mantenga el **TsgcDiagramDocument** o su JSON en el modelo de vista y reaccione a **OnChange** y **OnSelectionChange** de la vista.

Véase también

- Descripción general de .NET
- Diferencias de .NET respecto a Delphi
- TsgcDiagramDataBinding
- Enlace de datos

Renderizado sin ventana en .NET

El documento no necesita ventana. Un servicio, una prueba o un informe puede construir o cargar un diagrama y guardarlo como JSON, SVG y PNG en cualquier plataforma, Linux incluido.

- **TsgcDiagramDocument** se crea con `new TsgcDiagramDocument()`. Los conectores se enrutan cuando termina la transacción o actualización más externa.
- **TsgcDiagramRenderer** escribe el diagrama completo con un margen (20 por defecto) a una escala: **SaveToSVGFile** y **SaveToSVGStream** con esegece.sgcDiagram, **SaveToPNGFile** y **SaveToPNGStream** con esegece.sgcDiagram.Rendering.
- **SaveToJSON**, **LoadFromJSON**, **SaveToFile**, **LoadFromFile**, **SaveToStream** y **LoadFromStream** del documento leen y escriben el formato de archivo de Delphi.

Ejemplo

Diseño automático

En la edición Registered un diagrama construido en código se puede ordenar antes de guardarlo. **Execute** es un paso de deshacer y vuelve a enrutar los conectores.

Notas

- El tamaño de la imagen son los límites del contenido más el margen, por la escala; una escala de 2 escribe una imagen del doble de ancho y de alto.
- **Theme** del documento elige los colores claros u oscuros de la exportación.
- **Painter.SetFontFile** del renderer toma un archivo de fuente TrueType normal y uno en negrita, para un servidor sin la fuente por defecto.
- Los controles también tienen **SaveToPNG** y **SaveToSVG**.

Véase también

- Descripción general de .NET
- Clases de .NET
- Inicio rápido sin ventana
- TsgcDiagramRenderer
- Diseño por capas
- Diseño en árbol

Demos de .NET

Las demos están en `.net\demos\Diagram`, una carpeta por demo con su proyecto, su `Program.cs` y un README.md. Construyen su interfaz en código, sin XAML. Cada demo tiene un conmutador `--test` que ejecuta una autocomprobación fuera de pantalla, escribe líneas PASS y FAIL y termina con el número de comprobaciones fallidas, y un conmutador `--snapshot` que escribe el diagrama en un archivo PNG.

Demo	Host	Muestra
FlowchartEditor	WinForms	Una paleta y una vista, la gemela de la demo FlowchartEditor de Delphi: arrastre formas desde la paleta, conéctelas con la herramienta de conector o desde un puerto, deshacer y rehacer, zoom, temas, el diseño por capas y archivos JSON, PNG y SVG.
FlowchartWpf	WPF	El mismo editor con **TsgcWpfDiagramPalette** y **TsgcWpfDiagramView**.
FlowchartAvalonia	Avalonia	El mismo editor con **TsgcAvaloniaDiagramPalette** y **TsgcAvaloniaDiagramView**, para Windows, Linux y macOS, con zoom por pellizco y desplazamiento con un dedo.
OrgChart	WinForms	Un `DataTable` de 60 empleados en un `DataGridView` y, mediante un **TsgcDiagramDataBinding**, un organigrama en una vista de solo lectura con el diseño en árbol; una edición en la cuadrícula actualiza el organigrama. Edición Registered.
NodeEditor	WinForms	Un editor de grafos de nodos: bloques con puertos de entrada y salida tipados, conectados solo cuando los tipos coinciden.
ConsoleRender	Consola	Sin ventana: construye en código un diagrama de flujo de tickets de soporte, lo ordena con el diseño por capas y escribe flowchart.png, flowchart.svg y flowchart.json, en Windows y Linux. La gemela de la demo Headless de Delphi.

La parte sin interfaz de las tres demos Flowchart (el diagrama de flujo de ejemplo, las acciones de la barra de herramientas y la autocomprobación) está en `FlowchartEditor_Shared.cs`, que compilan los proyectos WinForms, WPF y Avalonia, de modo que los tres hosts ejecutan las mismas comprobaciones.
En la edición Community las demos Flowchart conservan las posiciones aproximadas del ejemplo y ocultan el botón de diseño, ConsoleRender coloca las filas en código, y OrgChart muestra la tabla e indica que el organigrama necesita la edición Registered.

Compilar las demos

Dentro del árbol del producto las demos se compilan contra los proyectos de la biblioteca. El zip de las demos usa por defecto los paquetes NuGet de la edición Community (OrgChart usa los Registered). Compile una demo con los paquetes Registered estableciendo **SgcDiagramEdition**:

```
dotnet build FlowchartEditor\FlowchartEditor.csproj -p:SgcDiagramEdition=Registered
dotnet run --project ConsoleRender\ConsoleRender.csproj -- --test
```

Véase también

- Descripción general de .NET
- Instalación desde NuGet
- Inicio rápido WinForms
- Renderizado sin ventana en .NET

Solución de problemas

Problema	Causa y solución
Los conectores creados en el código no tienen ruta, o la ruta no sigue a un nodo movido.	Las rutas se calculan al final del par BeginUpdate y EndUpdate más externo, y solo mientras Routing.AutoRoute es True. Cierre cada BeginUpdate, o llame a RouteAll después de un cambio masivo hecho con AutoRoute desactivado.
Los conectores ortogonales cruzan los nodos.	En la edición Community, los conectores dgcOrthogonal son codos simples sin esquiva de obstáculos. La edición Registered enruta alrededor de los nodos.
Layers.Add genera EsgcDiagramError.	La edición Community permite una capa. Consulte Capas.
Un nodo de imagen muestra un cuadro tachado.	La clave de la imagen no está en Images, o los datos no son un PNG y ningún manejador de OnDecodeImage los decodificó.
Eliminar una imagen genera EsgcDiagramError.	Un nodo aún la usa en ImageKey. Cambie o elimine esos nodos primero.
Un arrastre desde el lado de un nodo seleccionado lo redimensiona en lugar de dibujar un conector.	Los tiradores de redimensión de un nodo seleccionado cubren sus puertos laterales. Haga clic primero en un espacio vacío, o establezca Tool en dgtConnector.
El editor de texto no se abre.	La vista es ReadOnly, el nodo o su capa es Locked, o un manejador de OnTextEditing estableció aAllow en False.
Pegar no hace nada después de copiar en otra aplicación.	El portapapeles de sgcDiagram es interno a la aplicación; no intercambia datos con otros programas.
Deshacer no hace nada después de un cambio hecho por código.	Un cambio fuera de BeginTransaction y EndTransaction borra las pilas de deshacer y rehacer. Envuelva los cambios de código en una transacción para poder deshacerlos.
No se pueden arrastrar formas desde la paleta en iOS o Android.	Las plataformas móviles no tienen servicio de arrastre. Toque un elemento de la paleta y luego toque la vista.
Tab no recorre los elementos en una vista FireMonkey.	Un formulario FireMonkey gestiona Tab antes que el control con foco y mueve el foco.
Establecer el Id de un elemento genera EsgcDiagramError.	Los Id son únicos en un documento. Elija un Id que ningún otro nodo o conector use.
Faltan las clases de diseño y de enlace de datos.	Forman parte de la edición Registered. Con la edición Community, sgcDiagram_Layout y sgcDiagram_Data son unidades vacías.

Consulte también

- Ediciones
- Plataformas

Index

Copyright © 2012-2026 eSeGeCe Software
info@esegece.com
www.esegece.com