sgcUI 2026.10

October 2026

Documentação para Delphi

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

Contents

sgcUI 26

Visão Geral do sgcUI 26
Instalação 27
Plataformas e Pacotes 29
Edições 31
Pacotes .NET 32

sgcPDF 33

Introdução 33
Instalação 35
Arquitetura 38
Início Rápido 41
TsgcPDFDocument 44
TsgcPDFEditor 48
TsgcPDFViewer 51
TsgcPDFTextExtractor 53
TsgcPDFExporter 55
TsgcPDFOptimizer 57
TsgcPDFSignatureHandler 59
TsgcPDFSignatureVerifier 61
Texto e Fontes 63
Imagens 65
Gráficos, Transparência e Sombreamentos 66
Tabelas 67
Códigos de Barras 69
Anotações 71
Marcadores 78
Camadas 80

Formulários 81
Criptografia 83
Assinaturas Digitais 85
PDF/A 87
Otimizando Arquivos PDF 89
Mesclar, Dividir e Extrair Páginas 90
Extração de Texto 91
Renderização e Exportação 93
Tratamento de Erros e Entradas Hostis 94
Arquitetura do Renderizador 96
TsgcPDFRendererRaster 98
TsgcPDFRendererGDI 101
Recursos de Renderização Suportados 103
TsgcPDFViewerControl 105
TsgcPDFThumbnailView 112
TsgcPDFOutlineView 113
TsgcPDFCommentsView 114
TsgcPDFViewerFMX 116
TsgcPDFCommentsViewFMX 120
Fontes OpenType 122
Scripts Complexos e Shaping 123
Texto em Chinês, Japonês e Coreano 125
Extração de Texto: Posições de Caracteres 126
Limites para Arquivos Não Confiáveis 128
Decodificadores de Imagem 129
Importação de TIFF 131
TsgcPDFImageExtractor 133
TsgcPDFCanvas 136
EMF para PDF 138
TsgcPDFVCLCanvasHelper 140
HTML para PDF 141

Códigos de Barras 2D e GS1-128 144
TsgcPDFBarcodeControl 146
TsgcPDFBarcodeFMX 150
TsgcPDFBarcodeSymbol 153
Linux 158
macOS, iOS e Android 160
C++Builder 161
Anexos 162
Faturas Factur-X e ZUGFeRD 165
Salvamento Incremental e Saída Compacta 167
Arquivos Grandes 169
FDF e XFDF 170
Valores e Aparências de Formulário 172
Assinaturas PAdES 174
Carimbos de Tempo e Validação de Longo Prazo 176
Verificação de Assinatura 178
TsgcPDFPAdESSigner (Ponte sgcSign) 180
Criptografia por Certificado 182
TsgcPDFRedactor 184
PDF Marcado 187
PDF/UA e PDF/A Nível A 190
Validador de Acessibilidade 191
Perguntas frequentes 193

sgcReports 195

Introduction 195
Instalação 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
Instalação 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 312
TsgcVCLChart 315
TsgcFMXChart 318
TsgcChartRenderer 321
TsgcChartInteraction 324
TsgcChartDataSource 331
TsgcChartDataSource_Manual 333
TsgcChartDataSource_DataSet 335

TsgcChartDataSource_JSON 337
TsgcChartDataSource_CSV 339
TsgcChartDataSource_Table 341
Bridge Overview 345
TsgcChartDataSource_MQTT 346
TsgcChartDataSource_WebSocket 348
TsgcChartDataSource_SSE 350
TsgcChartDataSource_Binance 352
TsgcChartSeries_Indicator 354
TsgcChartSeries_Oscillator 359
TsgcChartSeries_Volume 363
Panes 366
TsgcChartAggregator 370
TsgcChartDrawing 373
TsgcChartBuffer 378
Decimation 381
Canvas 383
Fonts 386
Indicators 389
.NET Overview 392
Início Rápido: WinForms 394
Início Rápido: WPF 395
Quick Start: Avalonia 396
Headless Rendering 398
Series Gallery 400
Bridge 403
.NET Differences from Delphi 406
Troubleshooting 409

sgcMaps 410

Introdução 410

Visão Geral 411
Instalação 414
Início Rápido VCL 416
Início Rápido FMX 418
Demos 420
01 Offline Map 421
02 Map Features 422
03 Services 424
04 Offline Services 426
05 Static Render 427
06 FireMonkey 429
07 Vector Tiles 430
TsgcVCLMap 431
TsgcFMXMap 436
Tiles, Camadas e Provedores 440
Tiles Vetoriais 443
Marcadores e Agrupamento 446
Formas e Rótulos 449
Mapas de Calor 451
GeoJSON, GPX e KML 453
Funções de Geometria 455
Renderização de Mapas Estáticos 457
Tiles Offline e o Baixador de Tiles 460
Leitor de OpenStreetMap 462
Índice de Geocodificação Offline 464
Grafos de Rota e Roteamento Offline 465
TsgcMapService 467
TsgcMapGeocoder 469
TsgcMapDirections 472
TsgcMapPlaces 475
TsgcMapElevation 477

TsgcMapTimeZone 479
TsgcMapRouteCalculator 481
TsgcMapOfflineRouter 484
TsgcMapOfflineGeocoder 486
TsgcMapGeoIndexBuilder 488
TsgcMapLocation 490
Ferramentas 493
GeoIndexBuilder 494
RouteGraphBuilder 496
Lazarus 497
Visão Geral do .NET 499
Instalação do .NET 503
Início Rápido: WinForms 505
Renderização Sem Interface Visual 507
Dados Offline do .NET 509
Mapeamento de Classes 512
Diferenças do .NET em Relação ao Delphi 515
Demos do .NET 517
Solução de Problemas 519

sgcDashboard 523

Introdução 523
Arquitetura 524
Edições 526
Instalação 527
Plataformas 529
Início Rápido VCL 533
Início Rápido FMX 535
Início Rápido C++Builder 536
Demos 537
TsgcVCLDashboard 538

TsgcVCLDashboardTile 541
TsgcFMXDashboard 543
TsgcFMXDashboardTile 546
TsgcDashboardKPI 548
TsgcVCLKPICard 551
TsgcFMXKPICard 553
TsgcDashboardSource 555
TsgcDashboardSource_WebSocket 557
TsgcDashboardSource_MQTT 559
TsgcDashboardSource_SSE 561
Layout 563
Temas 565
Persistência 566
Dados em Tempo Real 568
Imagens KPI sem Interface 570
Visão Geral do .NET 571
Instalação pelo NuGet 574
Início Rápido WinForms 576
Início Rápido WPF 577
Início Rápido Avalonia 578
Classes do .NET 579
Diferenças do .NET em relação ao Delphi 582
Vinculação de dados no .NET 585
Renderização sem Interface no .NET 587
Feeds em Tempo Real no .NET 589
Demos do .NET 591
Solução de Problemas 593

sgcGrid 595

Introdução 595
Arquitetura 596

Edições 598
Instalação 599
Plataformas 601
Desempenho 604
Início Rápido VCL 605
Início Rápido FMX 606
Início Rápido C++Builder 607
Demos 608
TsgcVCLGrid 609
TsgcFMXGrid 615
TsgcGridColumn 620
TsgcGridColumns 622
TsgcGridData 624
TsgcGridLineView 627
TsgcGridDataStore 631
TsgcGridRenderer 633
TsgcGridFormatRule 636
TsgcGridFormatRules 638
IsgcGridDataHost and IsgcUITable 639
TsgcGridSource_DataSet 641
TsgcGridFeedSource 643
TsgcGridFeedSource_WebSocket 645
TsgcGridFeedSource_MQTT 647
TsgcGridFeedSource_SSE 649
TsgcGridAIQuery 651
Atualizações em Tempo Real e Threads 653
Ordenação 655
Filtragem 656
Agrupamento 657
Resumos 660
Linhas de Árvore 662

Mestre Detalhe 665
Seleção e Teclado 669
Redimensionamento de Colunas 671
Colunas Congeladas 672
Flash ao Alterar 673
Formatação Condicional 674
Edição no Local 676
Modo Virtual 678
JSON e CSV 679
Exportação para Excel 680
Vinculação de DataSet 683
Feeds em Tempo Real 684
Pergunte à Sua Grade 686
Gráficos a Partir da Grade 688
Temas 689
Renderização Headless 690
Visão Geral do .NET 692
Instalação pelo NuGet 695
Início Rápido WinForms 697
Início Rápido WPF 698
Início Rápido Avalonia 699
Classes do .NET 700
Diferenças do .NET em Relação ao Delphi 704
Vinculação de dados no .NET 707
Renderização Headless no .NET 710
Vinculação de DataTable, JSON e CSV no .NET 711
Edição no Local no .NET 713
Feeds em Tempo Real no .NET 714
Pergunte à Sua Grade no .NET 716
Desempenho no .NET 717
Agrupamento e Resumos do .NET 719

Linhas de Árvore e Mestre Detalhe do .NET 722
Exportação para Excel do .NET 725
Demos do .NET 727
Solução de Problemas 729

sgcNavigation 731

Introdução 731
Arquitetura 733
Edições 734
Instalação 735
Plataformas 737
Início Rápido VCL 740
Início Rápido FMX 742
Início Rápido C++Builder 743
Demos 744
TsgcVCLNavigationBar 745
TsgcFMXNavigationBar 748
TsgcNavigationItem 751
TsgcNavigationItems 754
TsgcNavigationModel 756
Modo Sidebar 759
Modo Accordion 761
Recolhimento e Animação 762
Imagens e Glifos 764
Selos 766
Teclado e Foco 767
Action Links 769
Temas 770
Renderização Headless 771
Visão Geral do .NET 774
Instalação pelo NuGet 776

Início Rápido WinForms 777
Início Rápido WPF 778
Início Rápido Avalonia 779
Classes do .NET 780
Diferenças do .NET em relação ao Delphi 782
Vinculação de dados no .NET 784
Renderização Headless no .NET 786
Demos do .NET 788
Solução de Problemas 790

sgcToolbar 791

Introdução 791
Arquitetura 792
Edições 794
Instalação 795
Plataformas 797
Início rápido VCL 800
Início rápido FMX 802
Início rápido C++Builder 803
Demos 804
TsgcVCLToolbar 805
TsgcFMXToolbar 808
TsgcVCLStatusBar 811
TsgcFMXStatusBar 813
TsgcToolbarItem 816
TsgcToolbarItems 819
TsgcStatusBarPanel 820
TsgcStatusBarPanels 822
TsgcToolbarModel 823
TsgcStatusBarModel 826
Tipos de item 828

Alternadores e grupos de rádio 830
Menus suspensos 831
Excedentes 832
Action links 833
Imagens e glifos 834
Painéis de status 835
Progresso e indicadores 836
Atualizações thread-safe 837
Alça de redimensionamento 839
Teclado 840
Temas 841
Renderização headless 842
Visão geral do .NET 844
Instalação pelo NuGet 846
Início rápido WinForms 847
Início rápido WPF 848
Início rápido Avalonia 849
Classes do .NET 850
Diferenças do .NET em relação ao Delphi 853
Vinculação de dados no .NET 856
Renderização headless no .NET 858
Demos do .NET 859
Solução de problemas 861

sgcWizard 863

Introdução 863
Arquitetura 864
Edições 866
Instalação 867
Plataformas 869
Início Rápido VCL 872

Início Rápido FMX 874
Início Rápido C++Builder 876
Demos 877
TsgcVCLWizard 878
TsgcVCLWizardPage 881
TsgcFMXWizard 883
TsgcFMXWizardPage 886
TsgcVCLStepper 888
TsgcFMXStepper 890
TsgcWizardStep 892
TsgcWizardSteps 894
TsgcStepperModel 895
Navegação e Páginas Ignoradas 897
Validação 899
Finish e Cancel 900
Teclado 901
Estados das Etapas 902
Layout do Stepper 904
Legendas dos Botões e Localização 906
Temas 907
Tempo de Design 908
Renderização Headless 909
Visão Geral do .NET 911
Instalação via NuGet 913
Início Rápido WinForms 914
Início Rápido WPF 915
Início Rápido Avalonia 916
Classes do .NET 917
Diferenças do .NET em relação ao Delphi 920
Vinculação de dados no .NET 922
Renderização Headless no .NET 924

Demos do .NET 926
Solução de Problemas 928

sgcPivotGrid 929

Introdução 929
Arquitetura 931
Edições 933
Instalação 934
Plataformas 936
Início rápido VCL 939
Início rápido FMX 941
Início rápido C++Builder 943
Início rápido DataSet 944
Início rápido headless 946
Demos 948
TsgcVCLPivotGrid 949
TsgcFMXPivotGrid 953
TsgcVCLPivotFieldList 956
TsgcFMXPivotFieldList 958
TsgcPivotEngine 960
TsgcPivotField 964
TsgcPivotCalculatedField 966
TsgcPivotSource_DataSet 967
TsgcPivotSource_Table 968
TsgcPivotGridRenderer 970
TsgcPivotFormatRule 972
Áreas 974
Agregações 976
Agrupamento de datas 978
Totais 980
Filtros 982

Top N 985
Mostrar valores como 986
Campos calculados 988
Persistência do layout 990
Exportação 992
Expandir e recolher 994
Ordenação 996
Drill-down 998
Formatação condicional 1000
Interoperabilidade com gráficos 1002
Teclado e acessibilidade 1004
Desempenho 1006
TsgcPivotGridAIQuery 1008
Pergunte à sua tabela dinâmica 1012
Visão geral do .NET 1017
Instalação pelo NuGet 1019
Início rápido WinForms 1020
Início rápido WPF 1021
Início rápido Avalonia 1022
Classes do .NET 1023
Diferenças do .NET em relação ao Delphi 1026
Vinculação de dados no .NET 1028
Renderização sem interface no .NET 1030
Demos do .NET 1031
Solução de problemas 1032

sgcPlanner 1034

Introdução 1034
Arquitetura 1035
Edições 1038
Instalação 1040

Plataformas 1042
Início Rápido VCL 1046
Início Rápido FMX 1048
Início Rápido C++Builder 1049
Demos 1050
TsgcPlannerData 1052
TsgcVCLScheduler 1057
TsgcFMXScheduler 1061
TsgcVCLGantt 1064
TsgcFMXGantt 1068
TsgcVCLKanban 1072
TsgcFMXKanban 1076
TsgcPlannerSource_DataSet 1079
TsgcPlannerICalendar 1082
TsgcPlannerMSProject 1084
TsgcPlannerRenderer 1086
Dados Compartilhados 1089
Visões Sincronizadas 1091
Calendário de Trabalho 1093
Recorrência 1095
Escala de Tempo e Zoom 1098
Arrastar e Soltar 1100
Dependencies 1103
Agendamento Automático 1105
Caminho Crítico 1107
Linhas de Base 1109
Raias e Limites de WIP 1111
Vínculo com DataSet 1113
Persistência JSON 1115
iCalendar 1117
XML do MS Project 1119

Renderização sem Janela 1121
Temas 1123
Recursos por Edição 1125
Sincronização com o Google Agenda 1127
TsgcPlannerGoogleSync 1131
TsgcPlannerICalSubscription 1136
Visão geral do .NET 1140
Instalação pelo NuGet 1143
Início Rápido WinForms 1144
Início Rápido WPF 1145
Início Rápido Avalonia 1147
Classes do .NET 1148
Diferenças do .NET em relação ao Delphi 1151
Dados e agendamento no .NET 1153
Vínculo de dados no .NET 1154
JSON, iCalendar e MS Project no .NET 1156
Renderização sem janela no .NET 1157
Sincronização com o Google Agenda e iCalendar no .NET 1158
Demos do .NET 1160
Solução de Problemas 1162

sgcEditors 1165

Introdução 1165
Arquitetura 1167
Controles 1170
Instalação 1172
Plataformas 1174
Início rápido VCL 1177
Início rápido FMX 1179
Início rápido C++Builder 1181
Início rápido data-aware 1182

Calendário mensal 1184
Seletor de data 1188
Seletor de hora 1192
Seletor de data e hora 1195
Controle deslizante de intervalo 1198
Seletor de cor 1201
Token edit 1204
Avaliação 1207
Spin edit 1210
Seletor de data DB 1213
Seletor de hora DB 1215
Spin edit DB 1217
Avaliação DB 1219
Controle deslizante de intervalo DB 1221
TsgcEditorModel 1223
TsgcEditorLocale 1227
TsgcEditorFieldLink 1230
Entrada de texto e IME 1233
Teclado e acessibilidade 1236
Temas 1238
Localização 1240
DPI 1242
Saída sem janela 1244
Edições 1246
Visão geral do .NET 1248
Instalação pelo NuGet 1250
Início rápido WinForms 1251
Início rápido WPF 1252
Início rápido Avalonia 1253
Classes do .NET 1254
Diferenças do .NET em relação ao Delphi 1257

Vinculação de dados no .NET 1259
Renderização sem janela no .NET 1260
Demos do .NET 1261
Solução de problemas 1262

sgcGauges 1264

Introdução 1264
Arquitetura 1265
Tipos de medidor 1267
Instalação 1269
Plataformas 1271
Início rápido VCL 1274
Início rápido FMX 1276
Início rápido sem janela 1278
Início rápido C++Builder 1279
Medidor radial 1280
Medidor linear 1282
Medidor digital 1284
LED 1286
Barra de LEDs 1288
Anel de progresso 1290
VU meter 1292
Relógio 1294
Bússola 1296
Anunciador 1298
TsgcCustomGauge 1300
TsgcGaugeScale e TsgcGaugeRange 1303
TsgcGaugePointer, TsgcGaugeThreshold e TsgcGaugeAnimation 1305
Escalas 1308
Faixas 1310
Ponteiros 1312

Limiares 1314
Animação 1316
Temas 1318
Saída sem janela 1320
Edições 1322
Visão geral da ponte 1324
Fonte MQTT 1327
Fonte WebSocket 1329
Fonte de eventos enviados pelo servidor 1331
Visão geral do .NET 1333
Instalação pelo NuGet 1336
Início rápido WinForms 1337
Início rápido WPF 1338
Início rápido Avalonia 1339
Classes do .NET 1340
Diferenças do .NET em relação ao Delphi 1343
Vinculação de dados no .NET 1344
Renderização sem janela no .NET 1345
Ponte sgcWebSockets no .NET 1346
Demos do .NET 1348
Solução de problemas 1349

sgcDiagram 1351

Introdução 1351
O que é o sgcDiagram 1353
Arquitetura 1354
Edições 1356
Plataformas 1357
Início Rápido VCL 1361
Início Rápido FMX 1363
Início Rápido C++Builder 1365

Início Rápido Sem Janela 1366
TsgcVCLDiagramView 1367
TsgcFMXDiagramView 1369
TsgcVCLDiagramPalette 1370
TsgcFMXDiagramPalette 1372
TsgcDiagramDataBinding 1373
Eventos 1375
TsgcDiagramDocument 1377
TsgcDiagramNode 1380
TsgcDiagramConnector 1382
TsgcDiagramPort 1384
TsgcDiagramStyle 1385
TsgcDiagramLayer 1386
TsgcDiagramController 1387
TsgcDiagramRenderer 1389
Formato JSON 1390
Layout em Camadas 1393
Layout em Árvore 1395
Vínculo de Dados 1397
Formas 1399
Conectores e Roteamento 1401
Portas e Fixação 1403
Edição e Teclado 1405
Ajuste e Guias 1407
Desfazer e Área de Transferência 1408
Camadas 1409
Serialização 1410
Exportação 1412
Temas 1413
Modo Visualizador 1414
Visão geral do .NET 1415

Instalação pelo NuGet 1418
Início rápido WinForms 1419
Início rápido WPF 1420
Início rápido Avalonia 1421
Classes do .NET 1422
Diferenças do .NET em relação ao Delphi 1424
Vinculação de dados no .NET 1426
Renderização sem janela no .NET 1428
Demos do .NET 1429
Solução de Problemas 1430

Index 1431

Visão Geral do sgcUI

O sgcUI é a suíte da eSeGeCe de componentes visuais e de documentos para Delphi e C++Builder. Os produtos compartilham uma única pasta de código-fonte, um único arquivo de include de versão, um único instalador e esta ajuda. Cada produto é licenciado separadamente, e o instalador instala apenas os produtos que a sua licença inclui.

Os produtos

Produto	Para que serve
sgcPDF	Uma biblioteca PDF nativa: cria, edita e renderiza documentos PDF, com texto e fontes, desenho, imagens, segurança, acessibilidade e controles de visualização. Veja sgcPDF.
sgcReports	Um motor de relatórios headless construído sobre o sgcPDF, para serviços do Windows, daemons Linux64 e endpoints HTTP, além de aplicações VCL e FMX. Veja sgcReports.
sgcCharts	Gráficos de dados ao vivo. As fontes de dados acrescentam pontos ou candles OHLC a um buffer circular, e o gráfico desenha um snapshot decimado, de modo que feeds rápidos continuam sendo renderizados com fluidez. Veja sgcCharts.
sgcMaps	Mapas interativos. Os controles de mapa desenham tiles raster e vetoriais de forma nativa, sem controle de navegador, e mostram sobre eles marcadores, rotas, formas, rótulos e mapas de calor. Veja sgcMaps.
sgcDashboard	Um dashboard formado por uma grade de tiles. Cada tile hospeda qualquer controle, como um gráfico, um mapa, um visualizador de PDF ou uma pré-visualização de relatório. Veja sgcDashboard.
sgcGrid	Uma grade de dados ao vivo. As linhas são atualizadas por chave a partir de qualquer thread, as células alteradas piscam e se esvaem, e rajadas de atualizações são agrupadas em repaints temporizados. Veja sgcGrid.
sgcNavigation	Uma barra de navegação que mostra as páginas de uma aplicação como uma lista de itens com ícones, legendas, badges e cabeçalhos de grupo. Veja sgcNavigation.
sgcToolbar	Uma barra de ferramentas e uma barra de status, com botões, botões de alternância, grupos de opções, botões suspensos, rótulos, separadores e espaçadores. Veja sgcToolbar.
sgcWizard	Assistentes passo a passo e indicadores de etapas: contêineres de páginas com uma faixa de etapas e uma barra de botões. Veja sgcWizard.
sgcUICore	A base compartilhada do sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard. Ela não é vendida nem selecionada isoladamente: o instalador a adiciona sempre que um desses produtos é instalado.

Para onde ir em seguida

- Instalação explica o instalador do sgcUI e como ele escolhe os produtos.
- Edições lista as edições Registered, Trial e Community de cada produto.
- Pacotes .NET lista os pacotes NuGet de cada produto.

Instalação

Todos os produtos sgcUI são instalados por um único instalador. Você não precisa baixar nem executar um instalador separado para cada produto.

Entrando na conta

O instalador Registered pede que você entre com o nome de usuário e a senha da sua conta eSeGeCe. Você também pode reutilizar a licença armazenada por uma instalação anterior ou ativar uma licença offline. O servidor de licenças retorna os produtos que a sua conta possui, e uma licença All-Access inclui todos os produtos.

Escolhendo os produtos

A página de componentes mostra uma caixa de seleção por produto. Um produto que a sua licença não inclui aparece desabilitado e desmarcado. Selecione os produtos que deseja instalar, junto com os demos e os arquivos de ajuda.
Alguns produtos precisam de outros, e o instalador os adiciona para você:

- Selecionar o sgcReports também instala o sgcPDF, o motor sobre o qual ele é construído.
- sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard são construídos sobre o sgcUICore, portanto o sgcUICore é instalado sempre que um deles é selecionado.

Integrações com o sgcWebSockets

O grupo **sgcWebSockets bridges** oferece sgcCharts for sgcWebSockets, sgcDashboard for sgcWebSockets, sgcGrid for sgcWebSockets e sgcReports HTML for sgcWebSockets. Uma ponte só está disponível quando o sgcWebSockets já está instalado na máquina e você possui o produto que ela estende. Caso contrário, ela aparece desabilitada.

Compilando os pacotes

Depois de copiar os arquivos, o instalador compila os pacotes na sua máquina para cada versão do Delphi e do C++Builder que você escolheu e registra os pacotes de design time na IDE. Os pacotes são compilados na ordem de dependência: primeiro o sgcUICore, depois os produtos construídos sobre ele, em seguida sgcPDF, sgcReports e sgcMaps, e por último as pontes sgcWebSockets.

Migrando de instalações autônomas

As versões anteriores do sgcPDF, sgcReports, sgcMaps e sgcCharts tinham instaladores próprios. Quando o instalador do sgcUI encontra uma dessas instalações, ele lista os library paths e os pacotes de design que a instalação antiga adicionou a cada IDE, e só os remove depois que você confirma essa página. As entradas dos caminhos de pesquisa e de navegação que ainda apontam para a pasta de um produto autônomo também são removidas, de modo que a IDE nunca encontre duas cópias de uma unit.

Edições Trial e Community

- O instalador Trial instala todos os produtos na edição trial, sem licença. Há um instalador Trial para Delphi e outro para C++Builder.
- O instalador Community instala sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard, junto com o sgcUICore. Ele também oferece o sgcReports quando um sgcPDF licenciado já está instalado, porque o sgcPDF não tem edição Community. O instalador Community nunca oferece as pontes sgcWebSockets. Veja Edições.

Detalhes de cada produto

Cada produto descreve seus pacotes, requisitos e arquivos instalados no seu próprio tópico Instalação:

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

Plataformas e Pacotes

O sgcUICore e cada produto de componentes sgcUI trazem um pacote de runtime independente de framework (sgc<Product>), um pacote de runtime VCL (sgc<Product>VCL), um pacote de runtime FireMonkey a partir do XE6 (sgc<Product>FMX), e um pacote de tempo de projeto VCL e outro FireMonkey (dclsgc<Product> e dclsgc<Product>FMX). O sgcUICore não tem pacote de tempo de projeto. O pacote neutro é compilado para Win32 e Win64, e a partir do Delphi 10.3 também para Linux64 e macOS. O pacote FireMonkey é compilado para macOS a partir do Delphi 10.3 e para Android e iOS a partir do Delphi 10.4. Os pacotes do C++Builder são compilados apenas para Windows. Cada produto lista seus pacotes, versões do IDE e plataformas de destino em seu próprio tópico:

Produto	Tópico
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: pacotes de runtime

C++Builder Win64 (toolchain clássica, C++Builder 12 e 13): os pacotes de runtime Win64 do sgcUI são vinculados pelo compilador Delphi, portanto as exceções tratadas dentro do código do sgcUI funcionam quando o seu aplicativo usa pacotes de runtime. O linker C++ Win64 (ilink64) ainda tem um defeito próprio: o código Delphi que ele vincula estaticamente em um aplicativo compilado com pacotes de runtime trava em `_DelphiExceptionHandler` na primeira vez que um bloco `except on E: <class>` captura uma exceção cuja classe está em um pacote de runtime.

- Quando Vincular com pacotes de runtime está habilitado (Opções do Projeto, Pacotes, Pacotes de Runtime), liste todos os pacotes de runtime do sgcUI que o seu projeto usa, para que nenhuma unidade do sgcUI seja vinculada ao executável.
- No .cbproj, cada um desses pacotes precisa estar tanto em PackageImports quanto em LinkPackageImports. Um pacote listado apenas em PackageImports não é vinculado.
- Quando Vincular com pacotes de runtime está desabilitado, tudo é vinculado estaticamente e funciona.
- Win32 e Win64x (Modern) não são afetados.

C++Builder 10.3 Rio, Win64: exceções em threads secundárias

Quando uma exceção Delphi é gerada em uma thread secundária, por exemplo dentro de um manipulador de evento que roda em uma thread do sgcWebSockets ou em qualquer TThread, e o código C++ dessa thread a captura, ou a propaga por uma função C++ que destrói um objeto local, como um UnicodeString temporário, o aplicati-

vo trava com uma violação de acesso. Isso é um defeito do runtime do C++Builder 10.3 Win64 (cc64260mt.dll). Acontece também sem código sgc, e o C++Builder 10.1, 10.2, 10.4, 11, 12 e 13 não são afetados. Soluções alternativas: use o C++Builder 10.4 ou posterior, ou, em código Win64 que roda em threads secundárias, evite gerar exceções, por exemplo TryStrToInt em vez de StrToInt.

C++Builder 13 Win64x: nomes de arquivo dos pacotes

Os pacotes de runtime Win64x (Modern) têm os mesmos nomes de arquivo dos pacotes Win64 clássicos, por exemplo sgcUICoreC13.bpl, e ambos são de 64 bits, portanto o Windows não consegue distingui-los. Um aplicativo Win64x que usa pacotes de runtime e encontra primeiro o bpl do Win64 não inicia (0xC0000139, ponto de entrada não encontrado). Coloque a pasta dos bpl Win64x (Bpl\Win64x) no PATH antes da pasta Win64, ou distribua os bpl Win64x na mesma pasta do executável. Os aplicativos vinculados sem pacotes de runtime não são afetados.

Delphi XE7 FireMonkey: FMX.Forms primeiro

Somente Delphi XE7: um programa FireMonkey pode terminar com "Runtime error 216" ao encerrar quando FMX.Types ou uma unidade FMX do sgcUI vem antes de FMX.Forms na ordem uses do programa e um controle com estilo foi usado (o FMX libera o serviço de seleção antes do estilo padrão). O XE6, o XE8 e as versões posteriores não são afetados. Solução alternativa: liste FMX.Forms como a primeira unidade na cláusula uses do .dpr, como o IDE faz em novos aplicativos FMX.

Capturar um formulário VCL com PaintTo

Para capturar em um bitmap com PaintTo um formulário VCL que contém controles do sgcUI, trave o canvas do bitmap em torno da chamada (`Bitmap.Canvas.Lock`, depois `Unlock`). PaintTo em um canvas destravado pode perder seu contexto de dispositivo quando um controle nativo do formulário trata uma mensagem durante a captura, e o bitmap sai em branco ou incompleto. Um formulário FireMonkey usa MakeScreenshot no lugar.

Veja Também

- Instalação
- Edições

Edições

Todos os produtos sgcUI são construídos a partir do mesmo código-fonte em cada edição. A edição apenas define quais unidades e recursos uma compilação inclui.

- **Registered**: todas as unidades dos produtos que a sua licença possui, incluindo as pontes sgcWebSockets.
- **Trial**: todos os produtos, para avaliação. As compilações trial são feitas com o define `SGC_TRIAL`.
- **Community**: uma edição gratuita dos produtos listados abaixo, compilada com o define `SGC_COMMUNITY`. Ela nunca inclui as pontes sgcWebSockets.

Edição Community por produto

Produto	Community	Não incluído no Community
sgcPDF	Não	O instalador Community não distribui o sgcPDF.
sgcReports	Sim, sobre um sgcPDF licenciado	A ponte sgcReports HTML para o sgcWebSockets. O sgcReports Community só é oferecido quando um sgcPDF licenciado já está instalado.
sgcCharts	Sim	Os recursos financeiros: indicadores, séries financeiras, o agregador e as ferramentas de desenho. A ponte sgcWebSockets.
sgcMaps	Não	O instalador Community não distribui o sgcMaps.
sgcDashboard	Sim	Arrastar e redimensionar tiles em tempo de execução, e SaveLayout e LoadLayout. A ponte sgcWebSockets.
sgcGrid	Sim	Edição de células no local. A ponte sgcWebSockets.
sgcNavigation	Sim	Os Action links dos itens de navegação.
sgcToolbar	Sim	O menu de excedentes e os Action links dos itens.
sgcWizard	Sim	Nada. A compilação Community tem o conjunto completo de recursos.
sgcUICore	Sim	Instalado com os produtos Community que precisam dele.

Cada produto explica suas edições com mais detalhes na sua própria ajuda, por exemplo Edições do sgcCharts.

Pacotes .NET

As versões .NET dos produtos sgcUI são distribuídas como pacotes NuGet. Cada produto tem um pacote principal, com o nome do produto, e pacotes complementares para um framework de UI ou uma integração. Adicione o pacote principal e os complementares de que a sua aplicação precisa.

Produto	Pacote principal	Pacotes complementares
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`. Esses pacotes são dependências dos produtos construídos sobre o sgcUICore, portanto o NuGet os restaura automaticamente.

Pacotes Community

Uma compilação Community de um pacote tem o mesmo id com o sufixo `.Community`, por exemplo `esegece.sgcGrid.Community`. Veja Edições para saber o que a edição Community de cada produto deixa de fora.

Introdução

sgcPDF é uma biblioteca Delphi para criar, abrir, editar, assinar, renderizar e visualizar documentos PDF. É escrita em Object Pascal puro e compilada no seu executável, portanto não há DLL de mecanismo PDF para implantar nem biblioteca de terceiros para licenciar.

Principais Recursos

- Crie documentos PDF a partir de código: texto com as 14 fontes padrão ou fontes TrueType e OpenType (CFF) incorporadas com subsetting, texto Unicode incluindo CJK, composição de script complexo (junção de caracteres árabes, ligaduras, texto da direita para a esquerda), linhas, curvas, imagens, transparência, gradientes e padrões.
- Desenho com uma API semelhante ao TCanvas (TsgcPDFCanvas), desenho de canvas EMF e VCL convertido em PDF vetorial, e HTML com CSS convertido em páginas PDF.
- Imagens: importação de JPEG, PNG, BMP e TIFF multipágina; decodificadores em Pascal puro para JPEG (baseline e progressivo), JPEG 2000, CCITT G3/G4 e JBIG2; extração de imagens para PNG, JPEG ou BMP.
- Tabelas com cabeçalhos, bordas, cores alternadas de linha e quebras de página automáticas.
- Códigos de barras: Code 128, GS1-128, Code 39, EAN-13, EAN-8, UPC-A, ITF, Codabar, QR Code (versões 1 a 40), DataMatrix e PDF417.
- Anotações, marcadores (outlines), camadas de conteúdo opcional e formulários interativos: criação, preenchimento, achatamento, caixas de lista de seleção múltipla, aparências de campo Unicode, importação e exportação de FDF e XFDF.
- Abra documentos existentes, edite-os, adicione marcas d'água, cabeçalhos, rodapés e números de página, combine, divida, extraia, gire e exclua páginas.
- Salvamento incremental que mantém as assinaturas válidas, saída compacta com object streams e cross-reference streams, e leitura sob demanda de arquivos maiores que 2 GB.
- Arquivos incorporados com relações PDF/A-3, e faturas eletrônicas Factur-X, ZUGFeRD e XRechnung.
- Extração de texto com a posição, a fonte e o tamanho de cada caractere.
- Criptografia com senhas (RC4 de 40 e 128 bits, AES-128, AES-256 revisão 6) ou com certificados de destinatário (criptografia de chave pública).
- Assinaturas digitais com arquivos PFX, o repositório de certificados do Windows ou qualquer assinante externo, PAdES (ETSI.CAdES.detached), carimbos de tempo de documento, DSS para validação de longo prazo, verificação de assinaturas, e PAdES de B-B a B-LTA por meio da ponte opcional sgcSign.
- Redação real que remove texto, pixels de imagem, gráficos e anotações das áreas marcadas ou dos resultados de busca.
- PDF Marcado com uma API de árvore de estrutura, saída PDF/UA-1 e PDF/A-1a, 2a e 3a, saída PDF/A-1b, 2b e 3b, e um validador de PDF/A e PDF/UA.
- Otimizador que compacta streams, remove objetos duplicados e não utilizados, e reduz a resolução das imagens.
- Renderização em Pascal puro de páginas para PNG e BMP em todas as plataformas, um renderizador GDI, e controles de visualização para VCL e FireMonkey com zoom, seleção, busca, preenchimento de formulários e impressão.
- Windows, Linux 64 bits, macOS, iOS e Android, com Delphi 7 a 13 e C++Builder 2007 a 13.

Requisitos

- Delphi 7 a Delphi 13, C++Builder 2007 a C++Builder 13.
- Windows de 32 e 64 bits para todos os recursos.
- Linux 64 bits, macOS, iOS e Android para o núcleo multiplataforma, o renderizador raster e o visualizador FireMonkey (Delphi XE5 ou posterior para FireMonkey).
- O renderizador GDI, TsgcPDFViewer, TsgcPDFExporter e os controles de visualização VCL usam o GDI do Windows. A assinatura com arquivos PFX ou o repositório de certificados do Windows e a criptografia por certificado usam a CryptoAPI do Windows (crypt32.dll, parte do Windows).

Componentes

Após a instalação, os componentes a seguir aparecem na página **sgcPDF** da Paleta de Ferramentas:

Componente	Pacote	Descrição
TsgcPDFDocument	Core	O documento PDF: cria páginas, desenha conteúdo, carrega, salva, criptografa.
TsgcPDFEditor	Core	Edição de arquivos PDF existentes: marcas d'água, números de página, combinação, formulários.
TsgcPDFTextExtractor	Core	Texto de uma página com as fontes e as posições dos caracteres.
TsgcPDFOptimizer	Core	Reduz o tamanho de um arquivo PDF.
TsgcPDFSignatureHandler	Core	Assina documentos PDF.
TsgcPDFSignatureVerifier	Core	Verifica as assinaturas de um documento PDF.
TsgcPDFRedactor	Core	Remove texto, imagens e gráficos de áreas das páginas.
TsgcPDFImageExtractor	Core	Lista e salva as imagens de um PDF.
TsgcPDFHTMLRenderer	Core	Converte HTML e CSS em páginas PDF.
TsgcPDFViewer	VCL	Uma janela nativa que exibe, pesquisa e imprime um PDF.
TsgcPDFExporter	VCL	Renderiza páginas em arquivos BMP, PNG ou JPEG e miniaturas.
TsgcPDFViewerControl	VCL	Controle de visualização VCL: zoom, seleção, busca, preenchimento de formulários, impressão.
TsgcPDFThumbnailView	VCL	Miniaturas de páginas para TsgcPDFViewerControl.
TsgcPDFOutlineView	VCL	Árvore de marcadores para TsgcPDFViewerControl.
TsgcPDFBarcodeControl	VCL	Controle de código de barras e QR Code: onze simbologias, rotação, saída PNG, SVG e PDF.
TsgcPDFViewerFMX	FMX	Controle de visualização FireMonkey para Windows, macOS, iOS, Android e Linux.
TsgcPDFBarcodeFMX	FMX	Controle FireMonkey de código de barras e QR Code.

Comece com Instalação e depois com o Início Rápido.

Instalação

Requisitos

- Delphi 7 a Delphi 13, ou C++Builder 2007 a C++Builder 13.
- Windows de 32 e 64 bits para todos os recursos. Linux 64 bits, macOS, iOS e Android para as unidades multiplataforma (Delphi XE5 ou posterior para FireMonkey), consulte Linux e macOS, iOS e Android.

Pacotes

O sgcPDF é dividido em três pacotes de tempo de execução, de modo que aplicativos e serviços que não usam a VCL ou o FireMonkey precisam apenas do núcleo:

Pacote	Requer	Conteúdo
sgcPDFDxx (núcleo em tempo de execução)	rtl	Todas as unidades independentes de plataforma: documento, editor, parser, gravador, fontes, shaping, imagens e decodificadores, formulários, anotações, criptografia, assinaturas, PDF/A, PDF marcado, redação, anexos, extração de texto e imagem, otimizador, TsgcPDFCanvas, HTML para PDF e o renderizador raster.
sgcPDFVCLDxx (tempo de execução VCL)	rtl, vcl, sgcPDFDxx	Unidades do Windows e da VCL: TsgcPDFRendererGDI, TsgcPDFViewer, TsgcPDFExporter, TsgcPDFViewerControl, TsgcPDFThumbnailView, TsgcPDFOutlineView, TsgcPDFBarcodeControl, TsgcPDFEMFPlayer e TsgcPDFVCLCanvasHelper.
sgcPDFFMXDxx (tempo de execução FMX, XE5 e posterior)	rtl, fmx, sgcPDFDxx	TsgcPDFViewerFMX, TsgcPDFBarcodeFMX e TsgcPDFRendererFMX.
dclsgcPDFDxx (tempo de design)	designide, rtl, vcl, sgcPDFDxx, sgcPDFVCLDxx	Registra os componentes principais e da VCL.
dclsgcPDFFMXDxx (tempo de design, XE5 e posterior)	designide, rtl, fmx, sgcPDFDxx, sgcPDFFMXDxx	Registra TsgcPDFViewerFMX e TsgcPDFBarcodeFMX.

xx é o sufixo da versão do Delphi, e os arquivos estão na pasta **Packages**:

Versão do Delphi	Sufixo	Pacotes FMX
Delphi 7	D7	Não
Delphi 2007	D2007	Não
Delphi 2009	D2009	Não
Delphi 2010	D2010	Não
Delphi XE a XE4	DXE ... DXE4	Não
Delphi XE5 a XE8	DXE5 ... DXE8	Sim

Versão do Delphi	Sufixo	Pacotes FMX
Delphi 10 Seattle	D10	Sim
Delphi 10.1 Berlin	D10_1	Sim
Delphi 10.2 Tokyo	D10_2	Sim
Delphi 10.3 Rio	D10_3	Sim
Delphi 10.4 Sydney	D10_4	Sim
Delphi 11 Alexandria	D11	Sim
Delphi 12 Athens	D12	Sim
Delphi 13	D13	Sim

Por exemplo, o Delphi 13 usa sgcPDFD13.dpk, sgcPDFVCLD13.dpk, sgcPDFFMXD13.dpk, dclsgcPDFD13.dpk e dclsgcPDFFMXD13.dpk. O C++Builder usa os pacotes .cbproj sgcPDFCxx, sgcPDFVCLCxx e dclsgcPDFCxx, consulte C++Builder.

Instalando os Pacotes

1. Abra e compile (ou construa) os pacotes de tempo de execução nesta ordem: **sgcPDFDxx.dpk**, **sgcPDFVCLDxx.dpk** e, se você usar o FireMonkey, **sgcPDFFMXDxx.dpk**.
2. Abra o pacote de tempo de design **dclsgcPDFDxx.dpk**, compile-o e selecione **Install**. Para o FireMonkey, faça o mesmo com **dclsgcPDFFMXDxx.dpk**.
3. Adicione a pasta **Source** do sgcPDF ao caminho de biblioteca: *Tools* > *Options* > *Language* > *Delphi* > *Library* > *Library Path* (em versões mais antigas *Tools* > *Options* > *Library*). Adicione-a para cada plataforma que você usar: Windows de 32 bits, Windows de 64 bits, Linux 64 bits, macOS, iOS e Android.

Verificando a Instalação

Crie um novo aplicativo VCL e abra a Paleta de Ferramentas. A página **sgcPDF** contém estes componentes (o TsgcPDFViewerFMX e o TsgcPDFBarcodeFMX aparecem em aplicativos FireMonkey):

Componente	Pacote	Descrição
TsgcPDFDocument	Core	O documento PDF: cria páginas, desenha conteúdo, carrega, salva, criptografa.
TsgcPDFEditor	Core	Edição de arquivos PDF existentes: marcas d'água, números de página, combinação, formulários.
TsgcPDFTextExtractor	Core	Texto de uma página com as fontes e as posições dos caracteres.
TsgcPDFOptimizer	Core	Reduz o tamanho de um arquivo PDF.
TsgcPDFSignatureHandler	Core	Assina documentos PDF.
TsgcPDFSignatureVerifier	Core	Verifica as assinaturas de um documento PDF.
TsgcPDFRedactor	Core	Remove texto, imagens e gráficos de áreas das páginas.

Componente	Pacote	Descrição
TsgcPDFImageExtractor	Core	Lista e salva as imagens de um PDF.
TsgcPDFHTMLRenderer	Core	Converte HTML e CSS em páginas PDF.
TsgcPDFViewer	VCL	Uma janela nativa que exibe, pesquisa e imprime um PDF.
TsgcPDFExporter	VCL	Renderiza páginas em arquivos BMP, PNG ou JPEG e miniaturas.
TsgcPDFViewerControl	VCL	Controle de visualização VCL: zoom, seleção, busca, preenchimento de formulários, impressão.
TsgcPDFThumbnailView	VCL	Miniaturas de páginas para TsgcPDFViewerControl.
TsgcPDFOutlineView	VCL	Árvore de marcadores para TsgcPDFViewerControl.
TsgcPDFBarcodeControl	VCL	Controle de código de barras e QR Code: onze simbologias, rotação, saída PNG, SVG e PDF.
TsgcPDFViewerFMX	FMX	Controle de visualização FireMonkey para Windows, macOS, iOS, Android e Linux.
TsgcPDFBarcodeFMX	FMX	Controle FireMonkey de código de barras e QR Code.

Arraste um componente para o formulário e compile o projeto.

Usando o sgcPDF Sem Instalar

Todos os componentes também podem ser criados em tempo de execução. Todo componente tem um construtor que não precisa de formulário, por exemplo `TsgcPDFDocument.Create`, portanto aplicativos de console, serviços, servidores web e DLLs precisam apenas da pasta **Source** no caminho de busca. A maioria dos exemplos desta ajuda são programas de console.

Arquitetura

O sgcPDF é dividido em unidades por recurso. Você adiciona à cláusula *uses* apenas as unidades cujos tipos utiliza.

Unidades Principais

Unidade	Conteúdo
sgcPDF_Document	TsgcPDFDocument, TsgcPDFPage, TsgcPDFDocumentInfo
sgcPDF_Editor	TsgcPDFEditor, TsgcPDFWatermarkOptions, sgcPDFMergeFiles, sgcPDFExtractPages, sgcPDFGetPageCount
sgcPDF_Fonts	TsgcPDFStandardFont (as 14 fontes padrão), sgcPDFMeasureTextWidth
sgcPDF_Fonts_TrueType	TsgcTrueTypeFont: análise, subsetting, incorporação e shaping de TrueType e OpenType (CFF)
sgcPDF_Shaping	TsgcPDFShaper: shaping GSUB/GPOS do OpenType, junção árabe, algoritmo bidirecional do Unicode
sgcPDF_Fonts_CFF, sgcPDF_Fonts_Type1	Programas de fonte CFF e Type 1 usados pelos renderizadores
sgcPDF_Fonts_CJK	TsgcPDFCIDFont, TsgcPDFCJKFontManager (baixo nível, somente Windows)
sgcPDF_ContentStream	TsgcPDFContentBuilder (operadores de desenho de página), TsgcPDFContentParser
sgcPDF_Graphics	TsgcPDFExtGState, TsgcPDFShading, TsgcPDFPattern, TsgcPDFGraphicsResources
sgcPDF_Canvas	TsgcPDFCanvas: desenho estilo TCanvas em páginas PDF
sgcPDF_HTML	TsgcPDFHTMLRenderer: HTML e CSS para PDF
sgcPDF_Images	TsgcPDFImage: carregamento de JPEG, PNG, BMP e TIFF
sgcPDF_JPEG, sgcPDF_JPX, sgcPDF_CCITT, sgcPDF_JBIG2	Decodificadores de imagem: JPEG, JPEG 2000, fax CCITT e JBIG2
sgcPDF_ColorSpaces	Avaliação de espaço de cor e função compartilhada pelos renderizadores e pelo extrator de imagens
sgcPDF_ImageExtract	TsgcPDFImageExtractor, sgcPDFEncodePNG, sgcPDFEncodeBMP
sgcPDF_Tables	TsgcPDFTable, TsgcPDFTableRow, TsgcPDFTableCell
sgcPDF_Barcodes	TsgcPDFBarcodeGenerator, TsgcPDFBarcodeOptions: códigos de barras lineares, QR Code, DataMatrix, PDF417, GS1-128
sgcPDF_Barcode_Symbol	TsgcPDFBarcodeSymbol: layout de código de barras independente de framework com saída SVG, PNG e PDF

Unidade	Conteúdo
sgcPDF_Annotations	TsgcPDFAnnotationManager e as classes de anotação
sgcPDF_Bookmarks	TsgcPDFBookmarks, TsgcPDFBookmarkItem
sgcPDF_Layers	TsgcPDFLayerManager, TsgcPDFLayer, TsgcPDFLayerGroup
sgcPDF_Forms	TsgcPDFAcroForm, TsgcPDFFormField
sgcPDF_Attachments	TsgcPDFAttachments: arquivos incorporados e anexos PDF/A-3
sgcPDF_Tagged	Árvore de estrutura de PDF Marcado
sgcPDF_Redaction	TsgcPDFRedactor
sgcPDF_Encryption	TsgcPDFEncryptionMethod, TsgcPDFEncryptionHandler, auxiliares de hash e cifra
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: o interpretador de conteúdo independente de plataforma
sgcPDF_Renderer_Canvas	TsgcPDFRenderCanvas: o backend de desenho abstrato
sgcPDF_Renderer_Fonts	Programas de fonte e contornos de glifos para os renderizadores
sgcPDF_Renderer_Raster	TsgcPDFRendererRaster, TsgcPDFRasterCanvas, sgcPDFRasterAddFontDirectory
sgcPDF_Parser	TsgcPDFParser: o leitor de PDF de baixo nível
sgcPDF_Writer	TsgcPDFWriter: o gravador de PDF de baixo nível
sgcPDF_Filters	Filtros de stream (Flate, LZW, ASCII85, ASCIIHex, RunLength, CCITT, JBIG2, preditores) e sgcPDFMaxDecodedSize
sgcPDF_Types	Modelo de objetos PDF (TsgcPDFDictionary, TsgcPDFArray, TsgcPDFStream...), TsgcPDFRect, TBytes

Unidades VCL (Windows)

Unidade	Conteúdo
sgcPDF_Renderer_GDI	TsgcPDFRendererGDI, TsgcPDFGDICanvas: renderização em um contexto de dispositivo do Windows
sgcPDF_Export	TsgcPDFExporter, TsgcPDFExportOptions, sgcPDFPageToImage, sgcPDFToImages

Unidade	Conteúdo
sgcPDF_Viewer	TsgcPDFViewer
sgcPDF_ViewerVCL	TsgcPDFViewerControl, TsgcPDFThumbnailView, TsgcPDFOutlineView
sgcPDF_EMF	TsgcPDFEMFPlayer, sgcPDFEMFAddPage
sgcPDF_CanvasVCL	TsgcPDFVCLCanvasHelper
sgcPDF_Barcode_VCL	TsgcPDFBarcodeControl

Unidades FireMonkey (XE5 e posterior)

Unidade	Conteúdo
sgcPDF_Renderer_FMX	TsgcPDFRendererFMX, TsgcPDFFMXCanvas, sgcPDFBGRAToFMXBitmap
sgcPDF_ViewerFMX	TsgcPDFViewerFMX
sgcPDF_Barcode_FMX	TsgcPDFBarcodeFMX, TsgcPDFBarcodeTextSettingsFMX

A unidade opcional sgcPDF_Sign_sgcSign (assinaturas PAdES por meio do sgcSign) não faz parte dos pacotes. Adicione-a ao seu projeto quando a utilizar.

Coordenadas e Unidades

Todas as posições e tamanhos estão em pontos PDF (1/72 de polegada). A origem é o canto **inferior esquerdo** da página, e Y cresce para cima. Uma página A4 tem 595 x 842 pontos e uma página US Letter tem 612 x 792 pontos. Retângulos passados como *Left, Bottom, Right, Top* usam o mesmo espaço de coordenadas. As cores são informadas como componentes R, G, B entre 0 e 1. TsgcPDFCanvas é a exceção: usa uma origem superior esquerda e a unidade que você escolher, como um TCanvas da VCL.

Índices de Página

Os índices de página são baseados em zero: a primeira página é 0 e a última é `PageCount - 1`.

Strings

Métodos que desenham com as fontes padrão recebem `AnsiString` (codificação WinAnsi). Métodos que desenham com fontes TrueType, o texto extraído, o conteúdo de anotações, os títulos de marcadores e os valores de formulário usam `WideString`, portanto suportam qualquer caractere Unicode que a fonte contenha.

Início Rápido

Os exemplos a seguir são procedimentos completos extraídos de programas de console que foram compilados e executados. Adicione as unidades listadas em cada seção à sua cláusula *uses*.

Criar um 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 e Editar um PDF

Unidades: sgcPDF_Fonts, sgcPDF_Editor. TsgcPDFEditor carrega um documento, aplica alterações a cada página e o salva.

```
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;
```

Extrair 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 uma Página em uma Imagem

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;
```

Visualizar um PDF

Unidades: sgcPDF_Viewer. TsgcPDFViewer abre uma janela nativa. Em um programa de console, `RunMessageLoop` mantém a janela ativa até que seja fechada.

```
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;
```

Veja também

- Texto e Fontes
- Renderização e Exportação

TsgcPDFDocument

Unidade: sgcPDF_Document
A classe central do sgcPDF. Ela contém as páginas de um documento, desenha novo conteúdo, carrega arquivos existentes e salva o resultado, opcionalmente criptografado ou como PDF/A. Também dá acesso aos marcadores, às anotações, aos campos de formulário e às camadas do documento.

Criando e Carregando

Método	Descrição
Create / Create(AOwner)	Cria um documento vazio.
LoadFromFile(const AFileName: string)	Carrega um arquivo PDF. Chame SetPassword antes para arquivos criptografados.
LoadFromStream(AStream: TStream)	Carrega um PDF a partir de um stream.
LoadFromBytes(const AData: TBytes)	Carrega um PDF a partir da memória.
SaveToFile(const AFileName: string)	Grava o documento em um arquivo.
SaveToStream(AStream: TStream)	Grava o documento em um stream.

Páginas

Membro	Descrição
AddPage: TsgcPDFPage	Adiciona uma página A4 (595 x 842 pontos).
AddPage(AWidth, AHeight: Double): TsgcPDFPage	Adiciona uma página do tamanho especificado em pontos.
InsertPage(AIndex: Integer; AWidth, AHeight: Double): TsgcPDFPage	Insere uma página em uma posição.
DeletePage(AIndex: Integer)	Remove uma página.
GetPage(AIndex: Integer) / Pages[AIndex]	Retorna uma página (propriedade padrão).
PageCount: Integer	Número de páginas.
AppendDocument(ADoc: TsgcPDFDocument)	Anexa as páginas de outro documento.

Recursos

Método	Descrição
AddTrueTypeFont(const AFontName: WideString; ABold, AItalic: Boolean): Integer	Carrega uma fonte TrueType instalada e retorna seu índice de fonte para TsgcPDFPage.DrawTextTT.
AddImage(const AFileName: string): Integer	Carrega um arquivo JPEG, PNG ou BMP e retorna seu índice de imagem.
AddImageFromStream(AStream: TStream): Integer	O mesmo que AddImage, a partir de um stream.

Texto

Método	Descrição
ExtractText(APageIndex: Integer): AnsiString	Texto de uma página como AnsiString.
ExtractTextW(APageIndex: Integer): WideString	Texto de uma página como Unicode.
FindText(APageIndex: Integer; const AText: WideString): Boolean	True quando a página contém o texto.
GetPageContentStream(APageIndex: Integer): TBytes	O stream de conteúdo decodificado de uma página.

Segurança e Conformidade

Método	Descrição
SetEncryption(AMethod: TsgcPDFEncryptionMethod; const AUserPassword, AOwnerPassword: AnsiString)	Criptografa o documento quando ele é salvo.
SetPassword(const APassword: AnsiString)	Senha usada para abrir um arquivo criptografado. Chame-a antes de LoadFromFile.
SetPDFALevel(ALevel: TsgcPDFALevel)	Salva o documento como PDF/A-1b, PDF/A-2b ou PDF/A-3b.

Propriedades

Propriedade	Tipo	Descrição
Info	TsgcPDFDocumentInfo	Title, Author, Subject, Keywords, Creator e Producer.
Bookmarks	TsgcPDFBookmarks	Estrutura de tópicos do documento. Consulte Marcadores.
Annotations	TsgcPDFAnnotationManager	Anotações de todas as páginas. Consulte Anotações.
AcroForm	TsgcPDFAcroForm	Campos de formulário interativo. Consulte Formulários.
Layers	TsgcPDFLayerManager	Grupos de conteúdo opcional. Consulte Camadas.

Propriedade	Tipo	Descrição
GraphicsResources	TsgcPDFGraphicsResources	Estados de transparência, gradientes e padrões.
IsLoaded	Boolean	True depois que um documento é carregado.
Parser	TsgcPDFParser	O parser de um documento carregado, usado pelo extrator de texto, pelo validador e pelo renderizador.
PDFALevel	TsgcPDFALevel	O nível de PDF/A definido com SetPDFALevel.
Version (publicada)	AnsiString	Versão do PDF gravada no cabeçalho do arquivo.

TsgcPDFPage

Retornado por AddPage e GetPage. Seus métodos de desenho adicionam operadores ao conteúdo da página.

Método	Descrição
DrawText(const AText: AnsiString; X, Y: Double; AFont: TsgcPDFStandardFont; AFontSize: Double)	Texto com uma fonte padrão, começando no ponto de linha de base X, Y.
DrawTextColor(..., R, G, B: Double)	O mesmo, com uma cor de texto.
DrawTextTT(const AText: WideString; X, Y: Double; AFontIndex: Integer; AFontSize: Double)	Texto Unicode com uma fonte TrueType adicionada com AddTrueTypeFont.
DrawTextTTColor(..., R, G, B: Double)	O mesmo, com uma cor de texto.
DrawLine(X1, Y1, X2, Y2: Double; ALineWidth: Double = 1.0)	Linha. DrawLineColor adiciona uma cor.
DrawRectangle(X, Y, W, H: Double; ALineWidth: Double = 1.0)	Contorno de retângulo. DrawRectangleFill o preenche com uma cor.
DrawCircle(CX, CY, Radius: Double; ALineWidth: Double = 1.0)	Contorno de círculo. DrawCircleFill o preenche com uma cor.
DrawImageFromIndex(AImageIndex: Integer; X, Y, W, H: Double)	Desenha uma imagem adicionada com AddImage.
DrawImageFile(const AFileName: string; X, Y, W, H: Double)	Carrega e desenha uma imagem em uma única chamada.
SetOpacity(AOpacity: Double)	Opacidade (0 a 1) para as operações seguintes.
DrawRectangleTransparent(X, Y, W, H: Double; R, G, B, AOpacity: Double)	Retângulo preenchido semitransparente.
DrawLinearGradient(X, Y, W, H: Double; R0, G0, B0, R1, G1, B1: Double)	Retângulo preenchido com um gradiente linear.

Propriedades da página: **Width**, **Height**, **MediaBox**, **PageIndex**, **PageDict** e **Content**, o TsgcPDFContentBuilder que dá acesso a todos os operadores de desenho (consulte Gráficos).

Eventos

TsgcPDFDocument não tem eventos.

Exemplo

```
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;
```

Veja também

- Texto e Fontes
- Mesclar, Dividir e Extrair Páginas

TsgcPDFEditor

Unidade: sgcPDF_Editor
Edição de alto nível de um PDF existente. O editor envolve um TsgcPDFDocument (propriedade **Document**) e adiciona operações de uma única chamada: marcas d'água, cabeçalhos, rodapés, números de página, mesclagem, extração de páginas, preenchimento e achatamento de formulários.

Construtores

Construtor	Descrição
Create / Create(AOwner: TComponent)	Cria o editor com seu próprio documento.
Create(ADocument: TsgcPDFDocument)	Edita uma instância existente de TsgcPDFDocument.

Métodos

Método	Descrição
LoadFromFile / SaveToFile	Carrega e salva o documento.
AddBlankPage / AddBlankPage(AWidth, AHeight: Double)	Anexa uma página vazia.
DeletePage(AIndex: Integer)	Remove uma página.
RotatePage(AIndex: Integer; ADegrees: Integer)	Gira uma página. ADegrees deve ser 90, 180 ou 270.
AddText / AddTextColor(APageIndex, AText, X, Y, AFont, AFontSize [, R, G, B])	Desenha texto em uma página.
AddTransparentText(APageIndex, AText, X, Y, AFont, AFontSize, AOpacity)	Desenha texto semitransparente.
AddLine / AddRectangle / AddRectangleFill	Desenha formas em uma página.
AddImageToPage(APageIndex: Integer; const AImageFile: string; X, Y, W, H: Double)	Desenha um arquivo de imagem em uma página.
AddWatermark(APageIndex; const AOptions: TsgcPDFWatermarkOptions)	Texto girado e semitransparente em uma página.
AddWatermarkAllPages(const AOptions: TsgcPDFWatermarkOptions)	O mesmo em todas as páginas.
AddHeaderText / AddFooterText(const AText: AnsiString; AFont; AFontSize)	Texto na parte superior ou inferior de todas as páginas.
AddPageNumbers(AFont; AFontSize: Double; const AFormat: AnsiString)	Números de página em todas as páginas. AFormat usa dois %d: página atual e total de páginas.
AddTextAnnotation / AddLinkAnnotation / AddHighlightAnnotation	Adiciona anotações.

Método	Descrição
AddBookmark(const ATitle: WideString; APageIndex: Integer)	Adiciona um marcador de nível superior. AddBookmarkChild adiciona um marcador aninhado.
MergeFrom(const AFileName: string)	Anexa todas as páginas de outro PDF.
ExtractPages(AFromIndex, AToIndex: Integer; const AOutput-FileName: string)	Salva um intervalo de páginas em um novo arquivo.
ExtractText(APageIndex): AnsiString / ExtractTextUnicode(APageIndex): WideString / ExtractAllText	Extração de texto.
SearchText(APageIndex: Integer; const AText: WideString): Boolean	True quando a página contém o texto.
FillFormField(const AFieldName: AnsiString; const AValue: WideString)	Define o valor de um campo de formulário.
FlattenForm	Transforma os campos de formulário em conteúdo estático da página.
SetEncryption(AMethod; const AUserPassword, AOwnerPas-sword: AnsiString)	Criptografa a saída.
ConvertToPDFA(ALevel: TsgcPDFALevel)	Salva a saída como PDF/A.
SignDocument(const AOutputFile: string; const ACertSubject-Name, AReason, ALocation: WideString)	Salva e assina com um certificado do reposi-tório do Windows, localizado pelo nome do titular.

TsgcPDFWatermarkOptions

Record com os campos Text, Font, FontSize, Rotation (graus), Opacity (0 a 1) e R, G, B.

Funções Auxiliares

Função	Descrição
sgcPDFMergeFiles(const AInputFiles: array of string; const AOutputFi-le: string)	Mescla vários arquivos.
sgcPDFExtractPages(const AInputFile, AOutputFile: string; AFromPa-ge, AToPage: Integer)	Extrai um intervalo de páginas ba-seado em zero.
sgcPDFGetPageCount(const AFileName: string): Integer	Número de páginas de um arquivo.

Eventos

TsgcPDFEditor não tem eventos.

Exemplo

```
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;
```

Veja também

- Mesclar, Dividir e Extrair Páginas
- Formulários

TsgcPDFViewer

Unidade: sgcPDF_Viewer
Um visualizador de PDF implementado como uma janela nativa do Windows. Ele não depende da VCL, portanto funciona em aplicativos VCL (incorporado em qualquer controle com janela por meio de seu Handle) e em programas de console ou não visuais (como uma janela de nível superior). As páginas são desenhadas com o renderizador GDI.

Janela

Método	Descrição
CreateWindow(AParent: HWND; X, Y, W, H: Integer)	Cria o visualizador como uma janela filha, por exemplo de um TPanel.Handle.
CreateWindow(X, Y, W, H: Integer; const ATitle: AnsiString)	Cria uma janela de nível superior.
Show / Hide / Invalidate / DestroyWindow	Gerenciamento da janela.
RunMessageLoop	Executa um loop de mensagens até que a janela seja fechada. Use-o apenas em programas sem seu próprio loop de mensagens, como programas de console.

Documento e Navegação

Método	Descrição
LoadFromFile(const AFileName: string)	Abre um arquivo PDF.
LoadDocument(ADocument: TsgcPDFDocument; AOwns: Boolean = False)	Exibe um documento já em memória.
CloseDocument	Fecha o documento.
FirstPage / LastPage / NextPage / PrevPage / GotoPage(AIndex)	Navegação.
ZoomIn / ZoomOut / ZoomFit / ZoomFitPage / SetZoom(APercent: Double)	Zoom.
FindText(const AText: WideString): Boolean / FindNext / ClearSearch	Pesquisa de texto.
Print / PrintPage(APageIndex: Integer)	Impressão.
GetPageText(APageIndex: Integer): WideString	Texto de uma página.
ExportPageToBitmap(APageIndex: Integer; const AFileName: string; AWidth, AHeight: Integer)	Renderiza uma página em um arquivo BMP.

A janela também responde ao teclado: Left e Page Up vão para a página anterior, Right e Page Down para a próxima, Home e End para a primeira e a última página, as teclas de mais e menos ajustam o zoom, e Up e Down rolam a tela.

Propriedades

Propriedade	Tipo	Descrição
ViewMode (publicada)	TsgcPDFViewerMode	vmSinglePage (padrão) ou vmContinuous.
BackColor (publicada)	COLORREF	Cor de fundo ao redor das páginas. Padrão $00C8C8C8.
ShowAnnotations (publicada)	Boolean	Desenha as anotações. Padrão True.
Handle	HWND	Identificador (handle) da janela.
CurrentPage	Integer	Índice baseado em zero da página atual.
PageCount	Integer	Número de páginas.
Zoom	Double	Zoom atual.
Document	TsgcPDFDocument	O documento sendo exibido.

Eventos

TsgcPDFViewer não tem eventos.

Exemplo

```
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;
```

Veja também

- Renderização e Exportação

TsgcPDFTextExtractor

Unidade: sgcPDF_TextExtract
Extrai o texto de uma página, mantendo a posição, o nome da fonte, o tamanho da fonte e a largura de cada fragmento de texto. Ele decodifica codificações padrão, Differences de fonte, mapas ToUnicode e fontes CID, de modo que o texto escrito com fontes TrueType e CJK é retornado como Unicode.

Construtores

Construtor	Descrição
Create(AParser: TsgcPDFParser; APageDict: TsgcPDFDictionary)	Prepara a extração de uma página. Passe TsgcPDFDocument.Parser e TsgcPDFPage.PageDict de um documento carregado.
Create(AOwner: TComponent)	Construtor de componente, usado pela IDE. Use o construtor acima para extrair texto em código.

Métodos

Método	Descrição
Extract	Analisa o conteúdo da página. Chame-o antes dos outros métodos.
GetText: WideString	O texto da página na ordem de leitura, uma linha por linha de texto.
GetTextWithPositions: WideString	O texto com as coordenadas de cada fragmento.
GetFragmentCount: Integer	Número de fragmentos.
GetFragment(AIndex: Integer): TsgcPDFTextFragment	Um fragmento: Text, X, Y, FontSize, FontName e Width.
FindText(const ASearchText: WideString; ACaseSensitive: Boolean = False): Boolean	Pesquisa o texto da página.

Funções Auxiliares

Função	Descrição
sgcPDFExtractPageText(AParser: TsgcPDFParser; APageIndex: Integer): WideString	Texto de uma página.
sgcPDFExtractAllText(AParser: TsgcPDFParser): WideString	Texto de todas as páginas.
sgcPDFFindText(AParser: TsgcPDFParser; APageIndex: Integer; const AText: WideString): Boolean	Pesquisa em uma página.

Um TsgcPDFParser usado diretamente deve ser carregado e depois analisado: chame `LoadFromFile` e depois `Parse`.

Eventos

TsgcPDFTextExtractor não tem eventos.

Exemplo

```
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;
```

Veja também

- Extração de Texto

TsgcPDFExporter

Unidade: sgcPDF_Export
Renderiza as páginas de um TsgcPDFDocument em imagens BMP, PNG ou JPEG usando o renderizador GDI (TsgcPDFRendererGDI).

Métodos

Método	Descrição
ExportPage(ADocument: TsgcPDFDocument; APageIndex: Integer; const AFileName: string)	Renderiza uma página com a propriedade Options.
ExportPage(ADocument; APageIndex; const AFileName: string; const AOptions: TsgcPDFExportOptions)	Renderiza uma página com opções explícitas.
ExportPageToStream(ADocument; APageIndex; AStream: TStream; const AOptions)	Renderiza uma página em um stream.
ExportAllPages(ADocument; const AOutputDir, AFilePrefix: string [; const AOptions])	Renderiza todas as páginas em uma pasta.
ExportPages(ADocument; AFromPage, AToPage: Integer; const AOutputDir, AFilePrefix: string; const AOptions)	Renderiza um intervalo de páginas.
ExportThumbnail(ADocument; APageIndex; const AFileName: string; AMaxWidth, AMaxHeight: Integer)	Renderiza uma página redimensionada para caber na caixa especificada.
ExportAllThumbnails(ADocument; const AOutputDir, AFilePrefix: string; AMaxWidth, AMaxHeight: Integer)	Miniaturas de todas as páginas.

TsgcPDFExportOptions

Campo	Padrão	Descrição
Format	efPNG	efBMP, efPNG ou efJPEG.
DPI	150	Resolução. 72 DPI renderiza um pixel por ponto.
JPEGQuality	85	Qualidade do JPEG, de 1 a 100.
BackgroundColor	$00FFFFFF	Cor de fundo da página (COLORREF).
AntiAlias	False	Suaviza a saída.

Obtenha os padrões com `sgcPDFDefaultExportOptions`.

Funções Auxiliares

Função	Descrição
sgcPDFPageToImage(const APDFFile, AImageFile: string; APageIndex: Integer; ADPI: Integer = 150)	Abre um arquivo e renderiza uma página.
sgcPDFToImages(const APDFFile, AOutputDir: string; AFormat: TsgcPDFExportFormat = efPNG; ADPI: Integer = 150)	Abre um arquivo e renderiza todas as páginas.

Eventos

TsgcPDFExporter não tem eventos.

Exemplo

```
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;
```

Veja também

- Renderização e Exportação

TsgcPDFOptimizer

Unidade: sgcPDF_Optimize
Reescreve um arquivo PDF para torná-lo menor: compacta streams não compactados, remove objetos duplicados e não referenciados e pode reduzir a resolução das imagens e remover conteúdo opcional como metadados, marcadores, anotações, JavaScript e miniaturas.

Métodos

Método	Descrição
Optimize(const AInputFile, AOutputFile: string)	Otimiza um arquivo.
OptimizeStream(AInputStream, AOutputStream: TStream)	Otimiza um stream.
GetSizeReduction: Int64	Bytes economizados pela última otimização.
GetLog: TStringList	Relatório da última otimização. Pertence ao otimizador; não o libere.

TsgcPDFOptimizeOptions

Campo	Padrão	Descrição
CompressStreams	True	Flate compacta streams não compactados.
RemoveDuplicateObjects	True	Mescla objetos idênticos.
RemoveUnusedObjects	True	Descarta objetos inacessíveis a partir do documento.
DownsampleImages	False	Reduz imagens acima de MaxImageDPI.
MaxImageDPI	150	Resolução alvo para a redução.
JPEGQuality	75	Qualidade das imagens recompactadas.
RemoveMetadata	False	Remove os metadados XMP.
RemoveBookmarks	False	Remove a estrutura de tópicos.
RemoveAnnotations	False	Remove as anotações.
RemoveJavaScript	False	Remove as ações de JavaScript.
FlattenForms	False	Achata os campos de formulário.
RemoveThumbnails	False	Remove as miniaturas de página incorporadas.

Obtenha os padrões com `sgcPDFDefaultOptimizeOptions`, ou otimize um arquivo com os padrões em uma única chamada com `sgcPDFOptimizeFile(const AInputFile, AOutputFile: string)`.

Eventos

TsgcPDFOptimizer não tem eventos.

Exemplo

```
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;
```

Veja também

- Otimizando Arquivos PDF

TsgcPDFSignatureHandler

Unidade: sgcPDF_Signatures
Assina documentos PDF com um certificado X.509 e sua chave privada. A assinatura é uma assinatura PKCS#7 (CMS) destacada (*/SubFilter /adbe.pkcs7.detached*) criada por meio da CryptoAPI do Windows, e é adicionada como uma atualização incremental, de modo que os bytes originais permaneçam inalterados.

Certificado

Método	Descrição
LoadCertificateFromPFX(const AFileName: string; const APassword: AnsiString)	Carrega um certificado e a chave privada de um arquivo PFX (PKCS#12).
LoadCertificateFromStore(const ASubjectName: WideString)	Localiza um certificado pelo nome do titular no repositório MY do usuário atual.
LoadCertificateFromThumbprint(const AThumbprint: AnsiString)	Localiza um certificado pela impressão digital SHA-1.
ListCertificates: TStringList	Certificados do repositório. O chamador libera a lista.
GetCertificateInfo: TsgcPDFCertificateInfo	SubjectName, IssuerName, SerialNumber, NotBefore, NotAfter e Thumbprint do certificado carregado.

Assinatura

Método	Descrição
SetField(const AName: AnsiString; APageIndex: Integer; ALeft, ABottom, ARight, ATop: Double)	Campo de assinatura visível em uma página.
SetInvisible	Assinatura invisível.
SignDocument(const AInputFile, AOutputFile: string)	Assina um arquivo.
SignDocumentStream(AInputStream, AOutputStream: TStream)	Assina um stream.

Propriedades

Propriedade	Tipo	Descrição
SignerName (publicada)	WideString	Nome do assinante gravado na assinatura. O mesmo que Name.
Reason (publicada)	WideString	Motivo da assinatura.
Location (publicada)	WideString	Local da assinatura.

Propriedade	Tipo	Descrição
ContactInfo (publicada)	AnsiString	Informações de contato.
HashAlgorithm (publicada)	TsgcPDFSignatureHashAlgorithm	shaSHA1, shaSHA256 (padrão), shaSHA384 ou shaSHA512.
Appearance	TsgcPDFSignatureAppearance	Aparência da assinatura visível: ShowName, ShowDate, ShowReason, ShowLocation, Text, FontName, FontSize, ImageData.
CertInfo	TsgcPDFCertificateInfo	Informações do certificado carregado.

Eventos

TsgcPDFSignatureHandler não tem eventos.

Exemplo

```
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;
```

Veja também

- TsgcPDFSignatureVerifier
- Assinaturas Digitais

TsgcPDFSignatureVerifier

Unidade: sgcPDF_Signatures
Verifica as assinaturas digitais de um documento PDF: confere se cada assinatura é criptograficamente válida, se os intervalos de bytes assinados ainda cobrem todo o arquivo e se o certificado do assinante encadeia com uma raiz confiável no repositório de certificados do Windows.

Métodos

Método	Descrição
LoadDocument(const AFileName: string)	Carrega um PDF assinado.
GetSignatureCount: Integer	Número de assinaturas.
VerifySignature(AIndex: Integer): Boolean	True quando a assinatura corresponde aos bytes assinados.
IsDocumentModified(AIndex: Integer): Boolean	True quando o arquivo foi alterado após essa assinatura.
IsSignatureTrusted(AIndex: Integer): Boolean	True quando a cadeia de certificados é confiável para o Windows.
GetSignatureInfo(AIndex: Integer): TsgcPDFCertificateInfo	Informações do certificado do assinante.

Um certificado de teste autoassinado é verificado como válido, mas não confiável.

Eventos

TsgcPDFSignatureVerifier não tem eventos.

Exemplo

```
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;
```

Veja também

- TsgcPDFSignatureHandler
- Assinaturas Digitais

Texto e Fontes

O sgcPDF desenha texto com dois tipos de fontes: as 14 fontes padrão do PDF, que todo leitor de PDF fornece, e fontes TrueType, que são incorporadas no arquivo.

Fontes Padrão

As fontes padrão são selecionadas com **TsgcPDFStandardFont** (unidade sgcPDF_Fonts): sfCourier, sfCourierBold, sfCourierOblique, sfCourierBoldOblique, sfHelvetica, sfHelveticaBold, sfHelveticaOblique, sfHelveticaBoldOblique, sfTimesRoman, sfTimesBold, sfTimesItalic, sfTimesBoldItalic, sfSymbol e sfZapfDingbats. Elas não são incorporadas, portanto mantêm os arquivos pequenos, mas cobrem apenas o conjunto de caracteres WinAnsi. Use-as com `TsgcPDFPage.DrawText` e `DrawTextColor`.
`sgcPDFMeasureTextWidth(AFont, AText, AFontSize)` retorna a largura de uma string em pontos, que é como você centraliza ou alinha o texto à direita.

TrueType e Unicode

`TsgcPDFDocument.AddTrueTypeFont(AFontName, ABold, AItalic)` carrega uma fonte do Windows instalada pelo nome de sua família e retorna um índice de fonte. O texto desenhado com `DrawTextTT` ou `DrawTextTTColor` é uma WideString, portanto qualquer caractere contido na fonte pode ser usado. Quando o documento é salvo, apenas os glifos realmente usados são incorporados (subsetting de fonte) junto com um mapa ToUnicode, de modo que o texto pode ser pesquisado, copiado e extraído.

```
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

O texto em chinês, japonês e coreano funciona da mesma forma: adicione uma fonte TrueType que contenha os caracteres e desenhe com DrawTextTT. A fonte é incorporada como uma fonte CID com codificação Identity-H. A

enumeração **TsgcPDFCJKFont** (unidade sgcPDF_Fonts_CJK) lista fontes CJK comuns do Windows e `sgcPDFCJKFontNameFromEnum` retorna seus nomes de família:

Valor	Fonte	Idioma
cjkMSGothic, cjkMSMincho, cjkMeiryo	MS Gothic, MS Mincho, Meiryo	Japonês
cjkSimSun, cjkSimHei, cjkMicrosoftYaHei	SimSun, SimHei, Microsoft YaHei	Chinês Simplificado
cjkMingLiU, cjkMicrosoftJhengHei	MingLiU, Microsoft JhengHei	Chinês 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;
```

A fonte deve estar instalada na máquina que cria o PDF. O leitor do PDF não precisa dela, pois os glifos são incorporados.

Veja também

- TsgcPDFDocument
- Extração de Texto
- PDF/A

Imagens

O sgcPDF incorpora imagens JPEG, PNG e BMP. Os dados JPEG são armazenados como estão (DCTDecode), sem recompactação. Os pixels PNG e BMP são compactados com Flate.
Adicione uma imagem uma vez com `TsgcPDFDocument.AddImage` (ou `AddImageFromStream`) e desenhe-a quantas vezes forem necessárias com `TsgcPDFPage.DrawImageFromIndex`. O arquivo armazena a imagem apenas uma vez. `DrawImageFile` carrega e desenha um arquivo em uma única chamada. X e Y são o canto inferior esquerdo da imagem, e W e H seu tamanho na página, em pontos.

```
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;
```

A classe **TsgcPDFImage** (unidade sgcPDF_Images) carrega e inspeciona uma imagem sem um documento: Width, Height, BitsPerComponent, ColorSpace, HasAlpha. Outros formatos, como TIFF, geram uma exceção.

Veja também

- TsgcPDFDocument
- Otimizando Arquivos PDF

Gráficos, Transparência e Sombreamentos

TsgcPDFPage tem auxiliares para formas comuns: DrawLine, DrawRectangle, DrawRectangleFill, DrawCircle, DrawCircleFill e suas variantes com cor. Para transparência, use `SetOpacity` (afeta as operações que a seguem) ou `DrawRectangleTransparent`. `DrawLinearGradient` preenche um retângulo com um gradiente de cor, desenhado como faixas finas de cor da primeira cor até a segunda.

O Construtor de Conteúdo

`TsgcPDFPage.Content` retorna o **TsgcPDFContentBuilder** da página (unidade sgcPDF_ContentStream). Ele expõe os operadores de desenho do PDF um a um, e tudo o que grava vai para o mesmo conteúdo de página que os auxiliares:

- Estado gráfico: SaveState, RestoreState, SetMatrix, SetLineWidth, SetLineCap, SetLineJoin, SetMiterLimit, SetDashPattern.
- Cores: SetStrokeColorRGB, SetFillColorRGB, SetStrokeColorCMYK, SetFillColorCMYK, SetStrokeGray, SetFillGray.
- Caminhos: 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.
- Outros: SetGraphicsState, DrawXObject, BeginMarkedContent, EndMarkedContent e AddOperator para gravar qualquer operador.

Sempre combine SaveState com RestoreState para que as alterações não vazem para desenhos posteriores.

```
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;
```

Veja também

- TsgcPDFDocument
- Camadas

Tabelas

TsgcPDFTable (unidade sgcPDF_Tables) organiza linhas e colunas, quebra o texto dentro das células e desenha bordas e fundos.

- Crie-a com o número de colunas e defina as larguras com SetColumnWidths, SetColumnWidth ou SetColumnWidthPercent.
- SetHeaders adiciona uma linha de cabeçalho. AddRowData adiciona uma linha de dados a partir de um array de strings. AddRow e AddHeaderRow retornam um TsgcPDFTableRow para controle total sobre suas células (TsgcPDFTableCell com Style, ColSpan e RowSpan).
- Estilo: SetDefaultFont, SetHeaderFont, SetHeaderBackground, SetAlternateRowColors, SetBorder, SetPadding, ou os records DefaultStyle e HeaderStyle.

Desenho

Método	Descrição
DrawOnPage(APage: TObject; AX, AY: Double)	Desenha a tabela inteira em um TsgcPDFPage. AX, AY é o canto superior esquerdo.
DrawOnPageWithOverflow(APage: TObject; AX, AY, AMaxHeight: Double): Integer	Desenha as linhas de cabeçalho e as linhas de dados que couberem em AMaxHeight, e retorna o índice da primeira linha que não coube.
DrawMultiPage(ADocument: TObject; AX, AY, AMaxHeight: Double; APageWidth, APageHeight: Double)	Adiciona uma página do tamanho especificado a um TsgcPDFDocument, desenha a tabela a partir de AX, AY e adiciona mais páginas conforme necessário, repetindo as linhas de cabeçalho em 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;
```

Veja também

- TsgcPDFDocument

Códigos de Barras

TsgcPDFBarcodeGenerator (unidade sgcPDF_Barcodes) desenha códigos de barras como gráficos vetoriais, de modo que são impressos com nitidez em qualquer tamanho. Simbologias suportadas (TsgcPDFBarcodeType): bcCode128, bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcQRCode, bcDataMatrix, bcPDF417 e bcGS1_128. As versões e níveis de correção de erro do QR Code, DataMatrix, PDF417 e GS1-128 são descritos em Códigos de Barras 2D e GS1-128.
O primeiro parâmetro dos métodos Draw é a página (TsgcPDFPage) ou seu **construtor de conteúdo**, `TsgcPDFPage.Content`. Qualquer outro objeto gera uma exceção. AX e AY são o canto inferior esquerdo.

Método	Descrição
DrawBarcode(APage: TObject; ABarcodeType; const AData: AnsiString; AX, AY: Double)	Desenha com as opções padrão.
DrawBarcode(APage: TObject; ABarcodeType; const AData: AnsiString; AX, AY: Double; const AOptions: TsgcPDFBarcodeOptions)	Desenha com opções personalizadas.
DrawQRCode(APage: TObject; const AData: AnsiString; AX, AY, AModuleSize: Double)	Desenha um QR Code cujos módulos têm AModuleSize pontos de largura.
EncodeLinear(ABarcodeType: TsgcPDFBarcodeType; const AData: AnsiString): TBytes	Retorna as larguras de barras e espaços em módulos de uma simbologia linear (Code 128, Code 39, EAN-13, EAN-8, UPC-A, ITF, Codabar e GS1-128), começando por uma barra: as mesmas larguras que DrawBarcode desenha. Os tipos 2D geram uma exceção, pois têm codificadores próprios, consulte Códigos de Barras 2D e GS1-128.

TsgcPDFBarcodeOptions

Obtenha os padrões com `sgcPDFDefaultBarcodeOptions`: BarWidth 1.0, Height 50, ShowText True, TextFont sfHelvetica, TextFontSize 8, QuietZone 10, primeiro plano preto e fundo branco (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 exibir um código de barras em um formulário, solte um TsgcPDFBarcodeControl (VCL) ou um TsgcPDFBarcodeFMX (FireMonkey) e defina Symbology e Text. Os dois controles oferecem suporte às onze simbologias, incluindo os símbolos 2D e o GS1-128 descritos em Códigos de Barras 2D e GS1-128, e desenham cada módulo em pixels inteiros. Eles hospedam um TsgcPDFBarcodeSymbol, que também funciona sem formulário, por exemplo em um serviço ou no Linux:

- SaveToPNG e SaveToSVG salvam o símbolo como imagem.
- DrawToPDF desenha o símbolo em uma página PDF como retângulos vetoriais, com os mesmos módulos, a mesma rotação, a mesma zona de silêncio e a mesma linha de texto do controle.
- Runs, ModuleAt e LayoutAxes retornam os módulos e o layout em pixels para o desenho personalizado.

Veja também

- TsgcPDFDocument
- Códigos de Barras 2D e GS1-128
- TsgcPDFBarcodeControl
- TsgcPDFBarcodeSymbol

Anotações

As anotações são gerenciadas por `TsgcPDFDocument.Annotations` (TsgcPDFAnnotationManager, unidade sgcPDF_Annotations). Cada método Add recebe o índice de página baseado em zero e retorna a nova anotação para que você possa definir sua cor (SetColor), Author, Contents, Opacity e Flags.

Método	Anotação
AddText(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AText: WideString)	Nota adesiva (TsgcPDFTextAnnotation, com Icon e Open).
AddLink(APageIndex; ALeft, ABottom, ARight, ATop: Double; ADestPage: Integer)	Link para outra página.
AddLinkURI(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AURI: AnsiString)	Link para um endereço web.
AddFreeText(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AText: WideString)	Caixa de texto (SetFont, SetTextColor, Alignment).
AddHighlight / AddUnderline / AddStrikeOut(APageIndex; const AQuadPoints: array of Double)	Marcação de texto. Cada grupo de 8 números é um quadrilátero.
AddStamp(APageIndex; ALeft, ABottom, ARight, ATop: Double; const AStampName: AnsiString)	Carimbo, como Approved ou Draft.
AddLine(APageIndex; AX1, AY1, AX2, AY2: Double)	Linha com StartStyle e EndStyle (por exemplo OpenArrow).
AddInk(APageIndex; const APoints: array of Double)	Desenho à mão livre a partir de pares X, Y.
AddSquare / AddCircle(APageIndex; ALeft, ABottom, ARight, ATop: Double)	Retângulo ou elipse (TsgcPDFShapeAnnotation, com SetInteriorColor).
AddReply(AParent: TsgcPDFAnnotation; const AText: WideString)	Anotação de texto que responde a AParent.

Quando um documento é carregado, suas anotações existentes são lidas no mesmo gerenciador como objetos tipados, de modo que `Count`, `GetAnnotationsForPage` e `Annotations` as listam. Elas podem ser alteradas e salvas novamente, veja Lendo anotações e Edição e salvamento abaixo.

```
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` e `AddCircle` adicionam um retângulo ou uma elipse inscrita no retângulo indicado e retornam uma **TsgcPDFShapeAnnotation**. SetColor define a cor da borda, SetInteriorColor a cor de preenchimento (não há preenchimento se não for chamado) e BorderWidth a largura da borda em pontos. As formas são salvas com um fluxo de aparência, de modo que todos os visualizadores as desenham da mesma maneira.

```
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;
```

Lendo anotações

LoadFromFile e LoadFromStream leem as anotações de todas as páginas em `Annotations`. Cada uma é criada como a classe do seu tipo, com suas propriedades preenchidas a partir do arquivo, de modo que o teste `is` habitual ou uma conversão de tipo dá acesso a elas:

Subtipo no arquivo	Classe e propriedades lidas
Text	TsgcPDFTextAnnotation: Icon, Open.
Link	TsgcPDFLinkAnnotation: DestPage, ou o endereço web.
FreeText	TsgcPDFFreeTextAnnotation: fonte, cor do texto, Alignment.
Highlight, Underline, StrikeOut	TsgcPDFHighlightAnnotation (AnnotType indica qual): QuadCount, QuadPoint.
Stamp	TsgcPDFStampAnnotation: StampName.

Subtipo no arquivo	Classe e propriedades lidas
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, a classe base.

Toda classe de anotação tem estes membros, tanto para anotações carregadas de um arquivo quanto para as novas:

Membro	Descrição
Author, Contents: WideString	Autor (/T) e texto (/Contents). O texto salvo como UTF-16, como o Acrobat e outros visualizadores gravam comentários com acentos ou em alfabetos não latinos, é decodificado para Unicode.
ModDate: TDateTime	Data de modificação (/M) como hora local, 0 quando não há.
Name: AnsiString	Nome da anotação (/NM).
InReplyTo: TsgcPDFAnnotation	A anotação à qual esta responde, ou com a qual está agrupada (/IRT). nil para uma anotação de nível superior.
ReplyType: AnsiString	Vazio ou R para uma resposta, Group para um membro de grupo (/RT).
IsReply: Boolean	True quando InReplyTo está definido e ReplyType não é Group.
BorderWidth: Double	Largura da borda ou da linha, em pontos.
HasColor, ColorR, ColorG, ColorB	A cor (/C), como lida do arquivo ou definida com SetColor. Os componentes vão de 0 a 1.
Modified: Boolean	True quando uma anotação carregada mudou desde que o documento foi carregado, ou desde o último ResetModified. Um indicador para a interface do usuário. Sempre False para uma anotação criada no código.
ChangedSinceLoad: Boolean	True quando uma anotação carregada difere do estado que tinha quando o documento foi carregado, de modo que o próximo salvamento a grava novamente. ResetModified não o altera. Sempre False para uma anotação criada no código.
MoveBy(ADX, ADY: Double)	Move a anotação ADX, ADY pontos: seu Rect e os pontos da linha, os pontos dos quadriláteros ou os traçados à tinta.
SourceObjectNumber: Integer	Número de objeto de uma anotação carregada no arquivo, 0 para uma criada no código.

A geometria de linhas, marcações de texto e tinta é lida com X1, Y1, X2 e Y2 (TsgcPDFLineAnnotation), QuadCount e QuadPoint(AIndex) com AIndex de 0 a QuadCount * 8 - 1 (TsgcPDFHighlightAnnotation), e PathCount, PathPointCount(APath) e 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;
```

Respostas

Uma resposta é uma anotação cujo `InReplyTo` aponta para outra anotação (a entrada /IRT do arquivo), como as respostas a um comentário no Acrobat. LoadFromFile vincula cada resposta ao seu pai, também quando o pai está em outra página. O gerenciador tem estes membros para respostas e para a própria lista:

Membro	Descrição
GetAnnotationsForPage(APageIndex: Integer): TList	As anotações de uma página, na ordem da lista. Libere a lista.
GetReplies(AAnnotation: TsgcPDFAnnotation): TList	As respostas diretas de uma anotação, na ordem da lista. Libere a lista.
AddReply(AParent: TsgcPDFAnnotation; const AText: WideString): TsgcPDFTextAnnotation	Adiciona uma anotação de texto que responde a AParent, na página dele e com o Rect dele.
Remove(AAnnotation: TsgcPDFAnnotation)	Remove e libera uma anotação, todas as anotações que respondem a ela ou estão agrupadas com ela (recursivamente) e seu popup.
LinkReplies	Resolve o /IRT das anotações carregadas para a anotação carregada com esse número de objeto. LoadFromFile o chama; chame-o você mesmo apenas depois de LoadFromPage.
Modified: Boolean	True quando uma anotação foi adicionada ou removida, ou uma anotação carregada mudou, desde que o documento foi carregado ou desde o último ResetModified. O indicador de "alterações não salvas" para a interface do usuário.
ChangedSinceLoad: Boolean	True quando uma anotação foi adicionada ou removida, ou uma anotação carregada difere do arquivo, desde que o documento foi carregado. ResetModified não o altera.
ResetModified	Limpa Modified, por exemplo depois de um salvamento. As edições são mantidas, e o próximo salvamento ainda as grava. Veja Controle de alterações.
MarkLoaded	Toma o estado atual como o estado do arquivo e limpa Modified e ChangedSinceLoad. O documento o chama depois de carregar. Veja Controle de alterações.

As respostas são gravadas com /IRT apontando para o pai por SaveToFile e por SaveIncremental, de modo que são lidas como respostas pelo sgcPDF e por outros visualizadores. AddReply não define Author nem ModDate, defina-os você mesmo.

Edição e salvamento

As propriedades de uma anotação carregada podem ser alteradas como as de uma nova: Contents, Author, Rect, a cor, Flags, ModDate e assim por diante, ou MoveBy. SaveToFile e SaveIncremental gravam as alterações.

- SaveIncremental grava novamente uma anotação alterada com o seu próprio número de objeto, depois dos bytes originais do arquivo. Anotações e respostas novas são adicionadas à sua página na mesma revisão.
- Quando ModDate não foi alterado pelo seu código, ele é definido com a hora atual.
- O fluxo de aparência do arquivo é mantido quando apenas Contents, Author, Name, ModDate, Rect, Flags ou as entradas de resposta mudam. Em outras alterações, carimbos e formas recebem uma nova aparência, e os demais tipos perdem a sua para que os visualizadores os desenhem novamente a partir das suas propriedades.
- Remove exclui uma anotação com suas respostas e seu popup, e o próximo salvamento grava a página sem eles.

Como SaveIncremental não altera os bytes originais, o intervalo de bytes de cada assinatura existente permanece o mesmo e o valor da assinatura continua válido. Se um validador aceita a nova revisão depende das permissões da assinatura. Uma assinatura de certificação com permissão DocMDP P=3, ou apenas assinaturas de aprovação, permitem adicionar e alterar comentários. Com P=1 ou P=2 o validador informa uma alteração não permitida. Um salvamento completo com SaveToFile grava o arquivo inteiro novamente e invalida as assinaturas existentes, portanto use-o apenas em documentos não assinados. Veja Salvamento Incremental e Assinaturas Digitais.

```
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;
```

Controle de alterações

Dois indicadores dizem o que mudou, e cada um tem a sua função.

- `Modified` é o indicador de "alterações não salvas" da interface do usuário. `Annotations.Modified` é True quando uma anotação foi adicionada ou removida, ou uma anotação carregada mudou, desde que o documento foi carregado ou desde o último ResetModified. A propriedade `Modified` de cada anotação indica quais anotações carregadas mudaram. Use-os para habilitar um botão Salvar, ou para perguntar ao usuário antes de fechar.
- `ChangedSinceLoad` compara com o arquivo. `Annotations.ChangedSinceLoad` é True quando uma anotação foi adicionada ou removida, ou uma anotação carregada difere do estado que tinha quando o documento foi carregado. SaveToFile e SaveIncremental usam a função `ChangedSinceLoad` de cada anotação para decidir quais anotações carregadas gravam com suas alterações.

`ResetModified` apenas limpa Modified. Chame-o depois de um salvamento, para que a interface do usuário mostre que não há mais nada a salvar. Ele não altera ChangedSinceLoad, portanto nunca perde uma edição: o próximo SaveToFile ou SaveIncremental ainda grava todas as alterações feitas desde que o documento foi carregado, junto com as mais recentes. Não é preciso carregar novamente o arquivo salvo, e o exemplo continua trabalhando no mesmo documento.
`MarkLoaded` toma o estado atual das anotações como o estado do arquivo e limpa os dois indicadores. LoadFromFile e LoadFromStream o chamam depois de ler as anotações, portanto os aplicativos raramente precisam dele. Para limpar o indicador de "alterações não salvas" depois de um salvamento, chame 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;
```

Renderizando anotações editadas

Para uso avançado, `TsgcPDFDocument.BuildPageAnnotations(APageIndex)` retorna as anotações de uma página como estão agora, com as edições incluídas, como um TsgcPDFArray para a propriedade Annotations de um renderizador. Quem chama libera o array. Os visualizadores o usam para desenhar páginas cujas anotações foram alteradas, portanto a maioria dos aplicativos nunca o chama.

Veja também

- TsgcPDFDocument
- Marcadores
- Salvamento Incremental
- TsgcPDFViewerControl
- TsgcPDFViewerFMX
- TsgcPDFCommentsView

Marcadores

Os marcadores (a estrutura de tópicos do documento) são gerenciados por `TsgcPDFDocument.Bookmarks` (TsgcPDF-Bookmarks, unidade sgcPDF_Bookmarks). `Add` cria uma entrada de nível superior e `TsgcPDFBookmarkItem.AddChild` uma aninhada. Cada item tem Title, PageIndex, Top, Left, Zoom, ColorR/G/B, Bold, Italic e 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;
```

Lendo Marcadores

Os marcadores de um documento carregado ficam disponíveis pela mesma propriedade.

```
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;
```

Veja também

- TsgcPDFDocument

- Anotações

Camadas

As camadas (grupos de conteúdo opcional) permitem que o leitor exiba ou oculte partes de uma página. Elas são gerenciadas por `TsgcPDFDocument.Layers` (TsgcPDFLayerManager, unidade sgcPDF_Layers).

- `AddLayer(AName, ADefaultState)` cria uma camada que está inicialmente ativada (lsOn) ou desativada (lsOff). TsgcPDFLayer também tem Intent, Visible, Printable e Locked.
- O conteúdo pertence a uma camada quando é gravado entre `BeginLayerMarkedContent(ALayer)` e `EndLayerMarkedContent`. Ambos retornam o texto do operador, que você adiciona à página com `Content.AddOperator`.
- `AddGroup` agrupa camadas no painel de camadas do leitor. `FindLayer`, `GetLayer`, `LayerCount` e `RemoveLayer` gerenciam a lista, que também é preenchida quando um documento é carregado.

```
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;
```

Veja também

- TsgcPDFDocument
- Gráficos

Formulários

Os formulários interativos (AcroForms) são gerenciados por `TsgcPDFDocument.AcroForm` (TsgcPDFAcroForm, unidade sgcPDF_Forms).

Criando Campos

Método	Campo
AddTextField(const AName: AnsiString; APageIndex: Integer; ALeft, ABottom, ARight, ATop: Double)	Caixa de texto.
AddCheckBox(...)	Caixa de seleção.
AddRadioGroup(const AName: AnsiString) and AddRadioButton(AGroup; APageIndex; ALeft, ABottom, ARight, ATop; const AExportValue)	Botões de opção.
AddComboBox(...; const AOptions: array of WideString)	Lista suspensa.
AddListBox(...; const AOptions: array of WideString)	Caixa de lista.
AddButton(...; const ACaption: WideString)	Botão de pressão.
AddSignatureField(...)	Campo de assinatura vazio.

Cada método retorna um TsgcPDFFormField com Value, DefaultValue, ReadOnly, Required, MaxLen, FontName, FontSize e Alignment. Defina `NeedAppearances` como True para que os leitores regenerem as aparências dos 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;
```

Preenchendo Campos

Quando um documento é carregado, seus campos são lidos em AcroForm. Use FillField, SetCheckBox, GetFieldValue, FindField, GetField e 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;
```

Achatamento

O achatamento desenha os valores atuais dos campos como conteúdo normal da página e remove os campos, de modo que os valores não podem mais ser editados. Use `TsgcPDFAcroForm.FlattenFields` ou `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;
```

Veja também

- TsgcPDFDocument
- TsgcPDFEditor

Criptografia

Chame `SetEncryption` antes de salvar para criptografar um documento com o manipulador de segurança padrão. **TsgcPDFEncryptionMethod** (unidade sgcPDF_Encryption) seleciona o algoritmo:

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

A senha de usuário é necessária para abrir o arquivo. A senha de proprietário concede acesso total. SetEncryption em TsgcPDFDocument e TsgcPDFEditor concede todas as permissões (imprimir, modificar, copiar, anotar, preencher formulários, extrair, montar e impressão de alta qualidade).

```
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;
```

Abrindo Arquivos Criptografados

Chame `SetPassword` antes de `LoadFromFile`. Tanto a senha de usuário quanto a de proprietário abrem o arquivo.

```
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;
```

Acesso de Baixo Nível

TsgcPDFEncryptionHandler implementa o próprio manipulador de segurança: `SetupEncryption` com um conjunto TsgcPDFPermissions, `Authenticate`, e as rotinas de criptografia e descriptografia para strings e streams. A unidade

também expõe sgcPDFMD5, sgcPDFSHA256, sgcPDFSHA384, sgcPDFSHA512, sgcPDFRC4, sgcPDFAESEncryptCBC e sgcPDFAESDecryptCBC. Erros nessa unidade geram **EsgcPDFEncryptionError**.

Veja também

- TsgcPDFDocument
- Assinaturas Digitais

Assinaturas Digitais

TsgcPDFSignatureHandler assina um PDF com um certificado que tem uma chave privada, carregado de um arquivo PFX, do repositório de certificados do Windows pelo nome do titular, ou pela impressão digital. A assinatura é uma assinatura PKCS#7 destacada (adbe.pkcs7.detached) anexada como uma atualização incremental, de modo que os bytes do arquivo original permanecem inalterados no início do arquivo assinado.

Assinando

```
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;
```

Chame `SetField` para um retângulo de assinatura visível, ou `SetInvisible` para uma assinatura sem aparência. `HashAlgorithm` seleciona SHA-1, SHA-256 (padrão), SHA-384 ou SHA-512.

Verificando

TsgcPDFSignatureVerifier informa, para cada assinatura, se ela é válida, se o documento foi modificado após a assinatura, e se o certificado é confiável para o 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;
```

Assinando com o Editor

`TsgcPDFEditor.SignDocument(AOutputFile, ACertSubjectName, AReason, ALocation)` salva e assina o documento editado com um certificado do repositório do Windows em uma única chamada.

Veja também

- Criptografia
- PDF/A

PDF/A

PDF/A é o padrão ISO para arquivamento de longo prazo. O sgcPDF grava os níveis de conformidade **b** (básico) do PDF/A-1, PDF/A-2 e PDF/A-3, selecionados com **TsgcPDFALevel** (unidade sgcPDF_PDFA): paA1b, paA2b e paA3b.
Quando `SetPDFALevel` é definido, o documento é salvo com metadados XMP que declaram o nível de conformidade e repetem os campos Info, uma intenção de saída sRGB e um ID de documento. Seu conteúdo também deve seguir as regras do PDF/A:

- Incorpore todas as fontes: desenhe o texto com fontes TrueType (AddTrueTypeFont e DrawTextTT), não com as fontes padrão.
- Não criptografe o documento.
- O PDF/A-1b não permite transparência, portanto evite SetOpacity e desenhos transparentes nesse nível.

```
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;
```

Validação

TsgcPDFAValidator verifica um documento carregado em relação a um nível: metadados XMP, incorporação de fontes, espaços de cor, transparência, criptografia, anotações, ações e estrutura do arquivo. `Errors` lista os problemas encontrados. É uma verificação rápida própria, não um substituto para um validador de conformidade completo como o 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)` define o nível em um documento aberto com o editor.

Veja também

- TsgcPDFDocument
- Texto e Fontes

Otimizando Arquivos PDF

TsgcPDFOptimizer reescreve um PDF para reduzir seu tamanho. Por padrão, compacta streams, mescla objetos duplicados e descarta objetos não utilizados. A redução de resolução de imagens e a remoção de metadados, marcadores, anotações, JavaScript e miniaturas são opcionais por meio de `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;
```

Um arquivo que já é compacto, como um arquivo gravado pelo próprio sgcPDF, pode não ficar menor. GetSizeReduction então retorna 0 e o log relata os tamanhos de entrada e saída.

Veja também

- Imagens

Mesclar, Dividir e Extrair Páginas

Auxiliares de Chamada Única

A unidade sgcPDF_Editor tem funções que operam diretamente sobre arquivos:

- `sgcPDFMergeFiles(AInputFiles, AOutputFile)` mescla arquivos na ordem informada.
- `sgcPDFExtractPages(AInputFile, AOutputFile, AFromPage, AToPage)` copia um intervalo de páginas baseado em zero e inclusivo para um novo arquivo (divisão).
- `sgcPDFGetPageCount(AFileName)` retorna o 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;
```

Com o Editor

TsgcPDFEditor combina operações de página em um documento aberto: MergeFrom, ExtractPages, DeletePage, RotatePage e 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;
```

Com um TsgcPDFDocument, use AppendDocument, InsertPage e DeletePage.

Veja também

- TsgcPDFEditor

Extração de Texto

O sgcPDF extrai texto dos content streams das páginas. Ele decodifica as codificações padrão (WinAnsi, MacRoman, Standard), Differences de fonte, mapas ToUnicode e fontes CID, de modo que o texto escrito com fontes TrueType ou CJK incorporadas é retornado como Unicode. Páginas digitalizadas que contêm apenas imagens não têm texto para extrair.

Texto Simples

`TsgcPDFDocument.ExtractTextW` retorna o texto de uma página e `FindText` o pesquisa.

```
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 com Posições

TsgcPDFTextExtractor retorna cada fragmento de texto com suas coordenadas, fonte e tamanho.

```
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;
```

Usando o Parser Diretamente

As funções sgcPDFExtractPageText, sgcPDFExtractAllText e sgcPDFFindText funcionam com um TsgcPDFParser. Chame `Parse` depois de carregar.

```
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;
```

Veja também

- TsgcPDFTextExtractor

Renderização e Exportação

As páginas são renderizadas com **TsgcPDFRendererGDI** (unidade sgcPDF_Renderer_GDI), que desenha o conteúdo da página (caminhos, texto, imagens, sombreamentos) em um contexto de dispositivo do Windows. Você raramente o usa diretamente:

- TsgcPDFExporter salva páginas como arquivos BMP, PNG ou JPEG, ou miniaturas.
- TsgcPDFViewer exibe páginas em uma janela.
- `TsgcPDFRendererGDI.RenderPage(ADC, AParser, APageDict, AWidth, AHeight)` desenha uma página em qualquer HDC, por exemplo um canvas de TBitmap ou uma impressora.

Exportando 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;
```

Visualizando

```
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;
```

Veja também

- TsgcPDFExporter
- TsgcPDFViewer

Tratamento de Erros e Entradas Hostis

O sgcPDF reporta erros gerando exceções. Carregar um arquivo danificado ou não suportado, um índice de página inválido, um certificado ausente ou uma imagem não suportada gera uma exceção cuja Message descreve o problema (por exemplo *PDF parse error: cannot find startxref* ou *sgcPDF: Invalid page index*). Erros de criptografia geram **EsgcPDFEncryptionError**. Envolva em try..except o carregamento de arquivos que você não 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;
```

Limites para Arquivos Não Confiáveis

Arquivos PDF podem ser criados para esgotar a memória ou a pilha, por exemplo com um pequeno stream compactado que se descompacta para gigabytes, ou com objetos profundamente aninhados. O parser e os decodificadores impõem limites e geram uma exceção quando um deles é excedido:

Limite	Valor	Descrição
sgcPDFMaxDecodedSize	268435456 (256 MB)	Tamanho máximo decodificado de um stream (Flate, LZW, ASCII85, ASCIIHex, RunLength). É uma variável global em sgcPDF_Filters, portanto você pode reduzi-la para serviços que processam uploads, ou aumentá-la para imagens legítimas muito grandes.
Aninhamento de objetos	512 níveis	Arrays e dicionários aninhados no parser.
Contagem de objetos	8388607	Maior número de objeto aceito a partir dos dados de referência cruzada.
Seções de referência cruzada	4096	Seções e streams xref encadeados.
Tamanho da imagem	2^{27} pixels, 1 GB de dados	Maior imagem que é decodificada.
Aninhamento de content stream	64 níveis	Profundidade de aninhamento aceita pelo parser de content stream.
Árvore de páginas	64 níveis, 1000000 nós	Profundidade e tamanho da árvore de páginas percorrida pelo extrator de texto.

Somente sgcPDFMaxDecodedSize pode ser alterado. Os demais limites são fixos. Defina sgcPDFMaxDecodedSize uma vez na inicialização, pois ele se aplica a todos os documentos do processo.

Veja também

- FAQ

Arquitetura do Renderizador

O sgcPDF desenha páginas PDF com um interpretador de conteúdo e vários backends de desenho. O interpretador lê o content stream da página, resolve fontes, imagens, espaços de cor, sombreamentos e anotações, e transforma cada coordenada em pixels do dispositivo. O backend só precisa pintar caminhos, rasters e glifos. Por isso, a mesma página tem a mesma aparência em qualquer backend.

Classe	Unidade	Função
TsgcPDFRenderer	sgcPDF_Renderer	O interpretador de conteúdo. Independente de plataforma, Pascal puro.
TsgcPDFRenderCanvas	sgcPDF_Renderer_Canvas	Backend de desenho abstrato: caminhos, recorte, imagens, rasters, camadas e texto do sistema.
TsgcPDFRasterCanvas	sgcPDF_Renderer_Raster	Rasterizador por software com anti-aliasing. Desenha em um buffer de pixels BGRA. Funciona em todas as plataformas.
TsgcPDFGDICanvas	sgcPDF_Renderer_GDI	Backend GDI do Windows. Desenha em qualquer contexto de dispositivo (tela, bitmap, impressora).
TsgcPDFFMXCanvas	sgcPDF_Renderer_FMX	O rasterizador por software com saída em bitmap FireMonkey e texto de sistema FMX. Delphi XE5 ou posterior.

Normalmente você não usa o interpretador diretamente. Cada backend tem uma classe de renderizador pronta que possui um interpretador e um canvas:

Renderizador	Saída	Plataformas
TsgcPDFRendererRaster	Arquivo ou stream PNG, arquivo BMP, pixels BGRA	Windows, Linux, macOS, iOS, Android
TsgcPDFRendererGDI	Contexto de dispositivo do Windows, arquivo BMP	Windows
TsgcPDFRendererFMX	FireMonkey TBitmap	Todas as plataformas FMX

Propriedades Comuns

Todas as classes de renderizador expõem o interpretador como `Renderer` e o backend como `Canvas`, além destas propriedades:

Propriedade	Descrição
RenderAnnotations: Boolean	Desenha as anotações da página e os campos de formulário depois do conteúdo da página. Padrão True.
AnnotationMode: TsgcPDFAnnotRenderMode	**armView** (padrão) ignora anotações com o flag NoView, **armPrint** desenha apenas anotações com o flag Print. Use armPrint para impressão e para imagens que devem corresponder a uma impressão. Não disponível em TsgcPDFRendererFMX, que sempre usa armView.

Escrevendo Seu Próprio Backend

Para desenhar em outro destino (uma biblioteca gráfica, um conversor de PDF para SVG, um gravador de testes), derive de **TsgcPDFRenderCanvas**, implemente seus métodos abstratos e passe-o para `TsgcPDFRenderer.Create(ACanvas)`. Depois chame `RenderPage(AParser, APageDict, AWidth, AHeight)`. O interpretador chama o canvas no espaço do dispositivo: as coordenadas são pixels com Y crescendo para baixo, as cores são bytes RGB e o alfa vai de 0 a 255. Os buffers de raster são BGRA de 32 bits, de cima para baixo e não pré-multiplicados.

Veja também

- TsgcPDFRendererRaster
- TsgcPDFRendererGDI
- Recursos de Renderização Suportados
- Renderização e Exportação

TsgcPDFRendererRaster

TsgcPDFRendererRaster (unidade sgcPDF_Renderer_Raster) renderiza páginas PDF em imagens sem GDI, VCL ou FMX. É escrito em Pascal puro, portanto o mesmo código funciona em um serviço do Windows, um servidor web Linux ou uma ferramenta de console no macOS. Use-o para criar pré-visualizações e miniaturas de páginas em um servidor, para converter PDF em PNG, ou para alimentar código de OCR ou processamento de imagem com pixels de página.

Métodos

Método	Descrição
RenderPageToPNG(AParser; APageDict; AWidth, AHeight: Integer; const AFileName: string)	Renderiza uma página em AWidth x AHeight pixels em um fundo branco e a salva como PNG.
RenderPageToBitmap(AParser; APageDict; AWidth, AHeight: Integer; const AFileName: string)	O mesmo, salvo como um arquivo BMP de 32 bits.
RenderPageToImage(AParser; APageDict; AWidth, AHeight: Integer)	Renderiza apenas no canvas. Leia os pixels de `Canvas.Bits` ou salve-os com os métodos do canvas.
RenderPage(AParser; APageDict; ARenderWidth, ARenderHeight: Integer; AOffsetX: Integer = 0; AOffsetY: Integer = 0)	Desenha sobre o conteúdo atual do canvas sem redimensioná-lo ou limpá-lo. Dimensione o canvas primeiro com `Canvas.SetSize`.
Render(AContentData: TBytes; AMediaBox: TsgcPDFRect; ...)	Renderiza um content stream bruto (sem recursos).

AParser é `TsgcPDFDocument.Parser` de um documento carregado, e APageDict é `TsgcPDFDocument.Pages[i].PageDict`. A rotação da página (/Rotate) é aplicada.

TsgcPDFRasterCanvas

A propriedade `Canvas` contém o raster da página:

Membro	Descrição
Width, Height	Tamanho do raster em pixels.
Bits: TBytes	Os pixels: BGRA, de cima para baixo, 4 * Width bytes por linha. O canal alfa é sempre 255.
SetSize(AWidth, AHeight)	Redimensiona o raster e o preenche de branco.
Clear(R, G, B: Byte)	Preenche o raster com uma cor.
SaveToPNGFile, SaveToPNGStream, SaveToBMPFile	Salva o raster.

Exemplo: Página para 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;
```

Exemplo: Todas as Páginas, Pixels e 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;
```

Fontes para Texto Não Incorporado

O texto em fontes incorporadas (TrueType, OpenType, CFF, Type 1 e Type 3) é desenhado a partir do programa de fonte no PDF. O texto em fontes que não são incorporadas, como as 14 fontes padrão, é desenhado com uma fonte TrueType do sistema de mesmo nome ou um substituto próximo. O renderizador varre estas pastas:

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

Chame `sgcPDFRasterAddFontDirectory(const ADirectory: string)` antes de renderizar para adicionar outra pasta (ela é varrida recursivamente). Em um servidor Linux mínimo ou um contêiner sem fontes, instale um pacote de fontes (por exemplo fonts-dejavu ou fonts-liberation) ou distribua suas próprias fontes e registre sua pasta.

Limites

O tamanho do raster é limitado a CS_CANVAS_MAX_RASTER pixels (67.108.864, por exemplo 8192 x 8192) para proteger o processo de alocações enormes. Divida saídas maiores em blocos (tiles) com `RenderPage` e os parâmetros de deslocamento.

Veja também

- Arquitetura do Renderizador
- Recursos de Renderização Suportados
- Linux
- TsgcPDFRendererGDI

TsgcPDFRendererGDI

TsgcPDFRendererGDI (unidade sgcPDF_Renderer_GDI) renderiza páginas PDF em um contexto de dispositivo do Windows. Usa o mesmo interpretador de conteúdo do renderizador raster e desenha por meio de **TsgcPDFGDICanvas**. TsgcPDFViewer, TsgcPDFViewerControl, TsgcPDFThumbnailView, TsgcPDFExporter e a impressão são construídos sobre ele. Somente Windows.

Métodos

Método	Descrição
RenderPage(ADC: HDC; AParser; APageDict; ARenderWidth, ARenderHeight: Integer; AOffsetX: Integer = 0; AOffsetY: Integer = 0)	Desenha uma página em um contexto de dispositivo: o canvas de um formulário ou controle (`Canvas.Handle`), um bitmap em memória ou uma impressora.
Render(ADC: HDC; AContentData: TBytes; AMediaBox: TsgcPDFRect; ...)	Desenha um content stream bruto.
RenderPageToBitmap(AParser; APageDict; AWidth, AHeight: Integer; const AFileName: string)	Renderiza uma página em um arquivo BMP.
RenderToBitmap(AContentData: TBytes; AMediaBox: TsgcPDFRect; AWidth, AHeight: Integer; const AFileName: string)	Renderiza um content stream bruto em um arquivo BMP.

Propriedades: `Renderer`, `Canvas`, `RenderAnnotations` e `AnnotationMode`, consulte Arquitetura do Renderizador.

Exemplo: Página para Arquivo Bitmap

```
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;
```

Exemplo: Pintar em um 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;
```

Veja também

- Arquitetura do Renderizador
- TsgcPDFRendererRaster
- TsgcPDFViewerControl

Recursos de Renderização Suportados

Esta página lista o que o interpretador de conteúdo desenha. Aplica-se a todos os renderizadores (raster, GDI e FMX) e aos visualizadores construídos sobre eles.

Gráficos

- Caminhos: linhas, curvas de Bézier, retângulos, preenchimento com as regras nonzero e even-odd, traço, recorte.
- Largura de linha, terminação (cap), junção (join), limite de meia-esquadria (miter limit) e padrões de traço (dash).
- Operadores de cor para todo espaço de cor (cs, CS, sc, SC, scn, SCN, g, rg, k e suas variantes de traço).
- Alfa constante de preenchimento e traço (ExtGState /ca e /CA).
- Modos de mesclagem: Normal e Multiply são compostos com exatidão. Outros modos de mesclagem são desenhados como Normal.
- Sombreamentos: baseado em função (tipo 1), axial (tipo 2), radial (tipo 3), malhas Gouraud de forma livre e de grade (tipos 4 e 5), patches Coons e de produto tensorial (tipos 6 e 7), usados com o operador sh e como padrões de sombreamento.
- Padrões de mosaico (tiling), coloridos e não coloridos.
- Form XObjects com sua matriz e caixa delimitadora.

Espaços de Cor

DeviceGray, DeviceRGB, DeviceCMYK, CalGray, CalRGB, Lab, ICCBased (por meio de seu alternado ou número de componentes), Indexed, Separation e DeviceN, com funções de transformação de matiz amostradas (tipo 0), exponenciais (tipo 2) e de calculadora PostScript (tipo 4). CMYK é convertido para RGB com uma fórmula simples, sem gerenciamento de cor ICC.

Imagens

- Image XObjects e imagens inline (BI, ID, EI).
- Filtros: DCTDecode (JPEG baseline e progressivo), JPXDecode (JPEG 2000), CCITTFaxDecode (Grupo 3 e Grupo 4), JBIG2Decode, FlateDecode e LZWDecode com preditores, RunLengthDecode, ASCII85Decode e ASCIIHexDecode. Consulte Decodificadores de Imagem.
- 1, 2, 4, 8 e 16 bits por componente, arrays /Decode, máscaras de imagem (stencil masks), máscaras de chave de cor, streams /Mask e máscaras suaves /SMask com /Matte.
- Posicionamento sob rotação, inclinação e inversões, com redução suave de escala.

Texto e Fontes

- Fontes TrueType incorporadas (FontFile2), fontes CFF e OpenType CFF (FontFile3), fontes Type 1 (FontFile), e fontes Type 3 (glifos desenhados como content streams).
- Fontes simples com codificações WinAnsi, MacRoman, Standard e /Differences, fontes simbólicas, e fontes compostas (Type0) com Identity-H, Identity-V e os CMaps CJK predefinidos.
- Escrita vertical (Identity-V e os CMaps verticais, com métricas /W2).
- Espaçamento de caractere e palavra, escala horizontal, elevação (rise), e os modos de renderização de texto (fill, stroke, clip, invisible).
- Fontes que não são incorporadas são desenhadas com uma fonte do sistema instalada de mesmo nome ou um substituto. Textos pequenos são superamostrados para permanecerem legíveis em resoluções baixas.

Anotações e Formulários

- Streams de aparência de anotações (/AP /N), com o estado /AS de caixas de seleção e botões de opção, os flags Hidden, NoView e Print, e o mapeamento da caixa delimitadora da aparência para o /Rect da anotação.
- Um desenho de reserva simples para tipos comuns de anotação sem stream de aparência.
- Quando o formulário define **/NeedAppearances** como true, campos de texto, campos de escolha e botões sem uma aparência normal são desenhados a partir dos valores de seus campos. `sgcPDFWidgetAppearanceGenerated(AParser, AWidget)` informa se um widget é desenhado dessa forma.
- Defina `RenderAnnotations := False` para desenhar apenas o conteúdo da página, e `AnnotationMode := armPrint` para desenhar o que uma impressora imprimiria.

Não Suportado

- Dicionários de máscara suave em ExtGState (/SMask diferente de /None) são ignorados, e grupos de transparência não são isolados nem recortados (knocked out).
- O conteúdo opcional (camadas) não é avaliado: o conteúdo de camadas ocultas é desenhado.
- Modos de mesclagem diferentes de Normal e Multiply são aproximados como Normal.

Veja também

- Arquitetura do Renderizador
- Decodificadores de Imagem
- Fontes OpenType

TsgcPDFViewerControl

TsgcPDFViewerControl (unidade sgcPDF_ViewerVCL) é um controle VCL que exibe documentos PDF dentro dos seus formulários. Arraste-o para um formulário, ou crie-o em tempo de execução, e carregue um arquivo. As páginas são renderizadas por TsgcPDFRendererGDI em segundo plano em um cache de páginas limitado, de modo que documentos grandes abrem rapidamente e rolam suavemente. Está na página de paleta **sgcPDF** junto com os controles complementares TsgcPDFThumbnailView, TsgcPDFOutlineView e TsgcPDFCommentsView.
Recursos: layout contínuo ou de página única, modos de zoom, navegação por teclado, roda do mouse e ferramenta de mão, seleção e cópia de texto, busca com realce, navegação por links, preenchimento de formulário interativo, revisão e edição de anotações e impressão. Compatível com alto DPI no Delphi 10.3 e posterior.

Propriedades Publicadas

Propriedade	Descrição
Document: TsgcPDFDocument	Exibe um TsgcPDFDocument que você possui. Deixe vazio para deixar o controle carregar os arquivos por conta própria.
FileName: string	Carrega um arquivo (também em tempo de design).
PageIndex: Integer	Página atual, baseada em zero.
Zoom: Double	Zoom em porcentagem. Defini-lo alterna ZoomMode para zmCustom.
ZoomMode: TsgcPDFViewerZoomMode	zmFitWidth (padrão), zmFitPage, zmActualSize ou zmCustom.
ViewMode: TsgcPDFViewerViewMode	pvmContinuous (padrão, todas as páginas uma abaixo da outra) ou pvmSinglePage.
PageGap: Integer	Espaço entre páginas em pixels. Padrão 10.
BackgroundColor: TColor	Cor ao redor das páginas. Padrão clAppWorkSpace.
ShowPageShadow: Boolean	Desenha uma sombra sob cada página. Padrão True.
CacheSize: Integer	Número de páginas renderizadas mantidas em memória. Padrão 12.
SelectionMode: TsgcPDFViewerSelectionMode	psmText (padrão) seleciona texto com o mouse, psmHand arrasta as páginas.
SelectionColor, SearchColor, SearchCurrentColor: TColor	Cores da seleção de texto, de todas as correspondências de busca e da correspondência atual.
FormFilling: Boolean	Permite que o usuário preencha campos de texto, caixas de seleção, botões de opção, caixas de lista e caixas de combinação. Padrão True.
OpenLinksInBrowser: Boolean	Abre links http, https e mailto com o aplicativo padrão. Padrão False.
AnnotationEditing: Boolean	Permite ao usuário selecionar, mover, redimensionar e excluir anotações, editar o texto delas e adicionar novas. Os links passam a ser selecionados em vez de se-

Propriedade	Descrição
	guidos. Definir como False limpa a seleção e mantém a ferramenta, porque as ferramentas também funcionam com AnnotationEditing em False. Padrão False.
AnnotationTool: TsgcPDFViewerAnnotationTool	Ferramenta para a próxima anotação colocada em uma página, com AnnotationEditing em True ou False: vatNone (padrão), vatNote, vatFreeText, vatHighlight, vatUnderline, vatStrikeOut, vatLine, vatArrow, vatSquare, vatCircle, vatInk ou vatStamp.
AnnotationAuthor: string	Autor das anotações e respostas que o usuário adiciona.
AnnotationColor: TColor	Cor das novas anotações. clDefault (padrão) usa a cor habitual de cada ferramenta.
AnnotationStampName: string	Nome do carimbo colocado por vatStamp, como Approved (padrão), Draft ou Confidential.
ShowAnnotationPopups: Boolean	Abre um popup quando se clica em uma anotação com conteúdo ou respostas. Padrão True.

Eventos

Evento	Descrição
OnPageChange: TNotifyEvent	A página atual mudou.
OnZoomChange: TNotifyEvent	O zoom mudou.
OnLinkClick(Sender; APageIndex: Integer; const AURI: string; var AHandled: Boolean)	Um link foi clicado. AURI está vazio para um link a uma página do documento. Defina AHandled como True para cancelar a ação padrão.
OnFieldChange(Sender; AField: TsgcPDFFormField)	O usuário alterou o valor de um campo de formulário.
OnSearchProgress(Sender; APageIndex, APageCount: Integer; var ACancel: Boolean)	Chamado para cada página enquanto FindText pesquisa. Defina ACancel para interromper.
OnAnnotationClick(Sender; AAnnotation: TsgcPDFAnnotation; var AHandled: Boolean)	Uma anotação com conteúdo ou respostas foi clicada. Defina AHandled como True para ignorar o popup.
OnAnnotationSelect(Sender; AAnnotation: TsgcPDFAnnotation)	A anotação selecionada mudou. AAnnotation é nil quando a seleção foi limpa.
OnAnnotationAdd(Sender; AAnnotation: TsgcPDFAnnotation)	O usuário adicionou uma anotação com uma ferramenta, ou uma resposta no popup.
OnAnnotationChange(Sender; AAnnotation: TsgcPDFAnnotation)	O usuário moveu, redimensionou ou editou uma anotação.
OnAnnotationDelete(Sender; AAnnotation: TsgcPDFAnnotation; var AAllow: Boolean)	Uma anotação está prestes a ser excluída com suas respostas. Defina AAllow como False para mantê-la.

Métodos Públicos

Método	Descrição
LoadFromFile, LoadFromStream, Close	Abre e fecha documentos.
GotoPage, NextPage, PrevPage, FirstPage, LastPage	Navegação.
ZoomIn, ZoomOut	Altera o zoom em um passo.
Print(AFromPage: Integer = 0; AToPage: Integer = -1; const ATitle: string = '')	Imprime um intervalo de páginas (baseado em zero; -1 é a última página) na impressora atual, com as anotações em modo de impressão.
RenderPageToBitmap(APageIndex: Integer; ABitmap: TBitmap; AZoomPercent: Double = 100; ADPI: Integer = 0)	Renderiza uma página em um bitmap VCL.
FindText(const AText: WideString; ACaseSensitive: Boolean = False; AWholeWord: Boolean = False; ABackward: Boolean = False): Boolean	Localiza e realça cada correspondência no documento e rola até a primeira.
FindNext, FindPrevious, ClearSearch, GetSearchMatch(AIndex)	Move entre correspondências. SearchResults é o número de correspondências, e SearchIndex a atual.
SelectText(APageIndex, AFirstChar, ACharCount), SelectPage, SelectAll, ClearSelection, CopyToClipboard	Seleção de texto. SelectedText e HasSelection leem a seleção.
PageText(APageIndex): TsgcPDFTextExtractor	Texto de uma página com posições de caracteres (pertence ao visualizador).
DeviceToPage, PageToDevice, GetPageRect, PageRotation	Converte entre pixels de cliente e o espaço de usuário PDF.
WidgetCount, GetWidget(AIndex), EditWidget	Acessa os widgets de campos de formulário e inicia a edição de um.
ClearCache, RenderVisiblePages, CachedPageCount	Controle de cache de páginas.
AnnotationAt(X, Y: Integer): TsgcPDFAnnotation	A anotação mais acima em um ponto do cliente, nil quando não há nenhuma. Respostas, popups, widgets de formulário e anotações ocultas são ignorados, e os links só são retornados com AnnotationEditing em True.
SelectedAnnotation, SelectAnnotation(AAnnotation), DeleteSelectedAnnotation: Boolean	A anotação selecionada. SelectAnnotation(nil) limpa a seleção. DeleteSelectedAnnotation funciona com AnnotationEditing em True, dispara OnAnnotationDelete e retorna True quando a anotação foi excluída.
ShowAnnotationPopup(AAnnotation), CloseAnnotationPopup	Abre o popup de uma anotação a partir do código, e o fecha.
GotoAnnotation(AAnnotation)	Rola a anotação para a área visível. Com AnnotationEditing em True ela também é selecionada.
AnnotationsChanged	Chame-o depois de alterar Document.Annotations no código. O visualizador é redesenhado e des-

Método	Descrição
	carta a seleção ou o popup de uma anotação removida.

Mouse e Teclado

- A roda do mouse rola, Ctrl + roda dá zoom.
- As setas, Page Up, Page Down, Espaço, Home e End rolam e mudam de página.
- Ctrl+A seleciona todo o texto do documento, Ctrl+C e Ctrl+Insert copiam a seleção.
- Tab move entre os campos de formulário durante o preenchimento.
- Um clique em uma anotação com conteúdo ou respostas abre o seu popup. Com AnnotationEditing em True, um clique seleciona uma anotação, arrastar a move, as alças a redimensionam, Delete a remove e as setas a movem (10 pontos com Shift).
- Esc durante o arraste de uma ferramenta, de um movimento ou de um redimensionamento cancela apenas esse arraste. Caso contrário, fecha o popup da anotação, limpa a seleção e cancela a ferramenta de anotação.

Exemplo: Visualizador com Miniaturas e 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;
```

Exemplo: 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;
```

Exemplo: Busca e Cópia

```
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;
```

Exemplo: Impressão

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

Anotações

O visualizador desenha as anotações de `Document.Annotations`. Clicar em uma anotação que tem conteúdo ou respostas abre um popup com seu autor, data, conteúdo e respostas. Trate OnAnnotationClick, ou defina ShowAnnotationPopups como False, para mostrar a sua própria interface.
Com **AnnotationEditing** em True o usuário também pode alterar as anotações:

- Um clique seleciona uma anotação e mostra suas alças. Arraste a anotação para movê-la, ou arraste uma alça para redimensioná-la.
- Delete remove a anotação selecionada com suas respostas. OnAnnotationDelete pode recusar a exclusão.
- As setas movem a anotação selecionada 1 ponto, ou 10 pontos com Shift.
- O texto do popup pode ser editado, e sua caixa Reply adiciona uma resposta escrita por AnnotationAuthor.
- Esc durante o arraste de uma ferramenta, de um movimento ou de um redimensionamento cancela apenas esse arraste. Caso contrário, fecha o popup, limpa a seleção e cancela a ferramenta atual.

Defina **AnnotationTool** para colocar uma nova anotação: clique em uma página, ou arraste sobre ela, e a anotação é criada com AnnotationAuthor e AnnotationColor. OnAnnotationAdd é disparado e a ferramenta volta para vatNone.

Ferramenta	Anotação colocada
vatNone	Nenhuma ferramenta (padrão). Os cliques selecionam anotações.
vatNote	Nota adesiva (TsgcPDFTextAnnotation).
vatFreeText	Caixa de texto (TsgcPDFFreeTextAnnotation).
vatHighlight, vatUnderline, vatStrikeOut	Marcação de texto (TsgcPDFHighlightAnnotation) sobre o texto que você seleciona com o mouse.
vatLine, vatArrow	Linha, ou linha com uma seta na ponta (TsgcPDFLineAnnotation).
vatSquare, vatCircle	Retângulo ou elipse (TsgcPDFShapeAnnotation).
vatInk	Desenho à mão livre (TsgcPDFInkAnnotation).
vatStamp	Carimbo com o nome AnnotationStampName (TsgcPDFStampAnnotation).

Salvar continua sendo tarefa do aplicativo. O visualizador só altera o documento na memória, e `Document.Annotations.Modified` indica se há algo a salvar. Chame `Document.SaveIncremental` para acrescentar as alterações e manter válidas as assinaturas existentes, ou `Document.SaveToFile` para um salvamento completo. Veja Anotações. Depois de alterar `Document.Annotations` no código, chame AnnotationsChanged. Para mostrar as anotações como uma lista de comentários ao lado do visualizador, vincule a ele um TsgcPDFCommentsView.

Exemplo: Anotações

```
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;
```

Preenchimento de Formulário

Com FormFilling True, clicar em um campo de texto ou de escolha abre um editor sobre o campo, e clicar em uma caixa de seleção ou botão de opção o alterna. Os valores dos campos são gravados no documento (TsgcPDFDocument.AcroForm) e a página é repintada. Trate OnFieldChange para reagir, e salve o documento com `Document.SaveToFile` para manter os valores preenchidos.

Veja também

- TsgcPDFThumbnailView
- TsgcPDFOutlineView
- TsgcPDFCommentsView
- TsgcPDFViewerFMX
- TsgcPDFViewer
- Anotações

TsgcPDFThumbnailView

TsgcPDFThumbnailView (unidade sgcPDF_ViewerVCL) mostra uma coluna rolável de miniaturas de página para um TsgcPDFViewerControl. Clique em uma miniatura para ir a essa página. A página atual é destacada, e as miniaturas são renderizadas sob demanda e armazenadas em cache.

Membro	Descrição
Viewer: TsgcPDFViewerControl	O visualizador cujo documento é exibido. As miniaturas acompanham seu documento e página atual.
ThumbnailWidth: Integer	Largura de uma miniatura em pixels. Padrão 120.
ItemCount	Número de miniaturas (páginas).
GetItemRect(AIndex): TRect, ItemAtPos(X, Y): Integer	Layout das miniaturas em coordenadas de cliente.
RenderVisibleThumbnails, CachedThumbnailCount	Controle de renderização e cache.

As propriedades VCL usuais (Align, Color, Font, PopupMenu...) e OnClick, OnMouseDown e OnResize são publicadas. Veja o exemplo em TsgcPDFViewerControl.

Veja também

- TsgcPDFViewerControl
- TsgcPDFOutlineView

TsgcPDFOutlineView

TsgcPDFOutlineView (unidade sgcPDF_ViewerVCL) é uma tree view que mostra os marcadores (outlines) do documento de um TsgcPDFViewerControl. Selecionar um nó rola o visualizador até o destino do marcador.

Membro	Descrição
Viewer: TsgcPDFViewer-Control	O visualizador a seguir. A árvore é reconstruída quando o visualizador carrega outro documento.
RebuildOutline	Recarrega a árvore a partir do documento.
NavigateToNode(ANode: TTreeNode)	Vai para o destino de um nó.

Descende de TCustomTreeView e publica as propriedades usuais de tree view (Items, Indent, ShowButtons, ShowLines, ShowRoot, HideSelection...) e os eventos OnClick e OnChange.

Veja também

- TsgcPDFViewerControl
- TsgcPDFCommentsView
- Marcadores

TsgcPDFCommentsView

TsgcPDFCommentsView (unidade sgcPDF_ViewerVCL) é uma vista de árvore que lista os comentários do documento de um TsgcPDFViewerControl, como o painel de comentários de um leitor de PDF. Todas as anotações são listadas, exceto campos de formulário, links e popups.

Membro	Descrição
Viewer: TsgcPDFViewerControl	O visualizador a seguir. A lista é reconstruída quando o visualizador carrega ou fecha um documento e quando suas anotações mudam.
GroupByPage: Boolean	Lista os comentários sob um nó por página, com o texto Page N (count). Quando False, os comentários formam uma lista única na ordem das páginas. Padrão True.
ShowReplies: Boolean	Mostra as respostas de um comentário como seus nós filhos. Quando False, apenas os comentários de nível superior são listados. Padrão True.
RebuildComments	Recarrega a lista a partir do documento do visualizador. A lista já acompanha o visualizador, e depois de alterar Document.Annotations no código o método AnnotationsChanged do visualizador também a reconstrói.
SelectedAnnotation: TsgcPDFAnnotation	Somente leitura. A anotação do nó selecionado, nil quando nenhum nó ou um nó de página está selecionado.

Cada nó mostra `Author, date: contents`. O conteúdo é exibido em uma linha, e textos longos são cortados em 120 caracteres. Quando o conteúdo está vazio, é mostrado o tipo da anotação, como Highlight ou Stamp (Approved). Uma anotação sem autor mostra (no author), e a data é omitida quando a anotação não tem nenhuma.

- Selecionar um nó rola o visualizador até a anotação com GotoAnnotation. Com AnnotationEditing do visualizador em True, a anotação também é selecionada no visualizador.
- Um clique duplo abre o popup da anotação.
- A tecla Delete exclui a anotação selecionada e suas respostas enquanto AnnotationEditing do visualizador for True. OnAnnotationDelete do visualizador pode recusar a exclusão.
- Quando o usuário seleciona uma anotação no visualizador, o nó dela é selecionado na lista.

Desce de TCustomTreeView e publica as propriedades habituais de uma vista de árvore (Items, Indent, ShowButtons, ShowLines, ShowRoot, HideSelection...) e os eventos OnClick e OnChange. A árvore é somente leitura, e HideSelection é False por padrão para que o comentário selecionado continue visível.

Exemplo: Painel de comentários

```
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;
```

Veja também

- TsgcPDFViewerControl
- TsgcPDFOutlineView
- TsgcPDFCommentsViewFMX
- Anotações

TsgcPDFViewerFMX

TsgcPDFViewerFMX (unidade sgcPDF_ViewerFMX) é um controle FireMonkey que exibe documentos PDF no Windows, macOS, iOS, Android e Linux (FMXLinux). As páginas são desenhadas pelo rasterizador por software em Pascal puro (TsgcPDFRendererRaster com saída em bitmap FMX) em um layout vertical contínuo, e mantidas em um cache de páginas limitado. Precisa do Delphi XE5 ou posterior e é instalado pelos pacotes FMX, na página de paleta **sgcPDF**.

Propriedades Publicadas

Propriedade	Descrição
Document: TsgcPDFDocument	Exibe um TsgcPDFDocument que você possui.
PageIndex: Integer	Página atual, baseada em zero.
Zoom: Single	Fator de zoom, 1 é 100 por cento. Defini-lo alterna ZoomMode para fzmCustom.
ZoomMode: TsgcPDFViewerFMXZoomMode	fzmFitWidth (padrão), fzmFitPage ou fzmCustom.
PageGap: Single	Espaço entre páginas.
BackgroundColor: TAlphaColor	Cor ao redor das páginas. Padrão $FF808080.
CacheSize: Integer	Número de páginas renderizadas mantidas em memória. Padrão 8.
OnPageChange, OnZoomChange, OnLayerChange: TNotifyEvent	Notificações de página, zoom e visibilidade de camadas.
OnLinkClick(Sender; APageIndex: Integer; const AURI: string; var AHandled: Boolean)	Um link foi tocado ou clicado. Links para páginas do documento são seguidos, a menos que AHandled seja definido.
AnnotationEditing: Boolean	Permite ao usuário selecionar, mover, redimensionar e excluir anotações, editar o texto delas e adicionar novas. Definir como False limpa a seleção e mantém a ferramenta, porque as ferramentas também funcionam com AnnotationEditing em False. Um popup aberto é aberto novamente no novo modo. Padrão False.
AnnotationTool: TsgcPDFViewerAnnotationTool	Ferramenta para a próxima anotação colocada em uma página, com AnnotationEditing em True ou False: vatNone (padrão), vatNote, vatFreeText, vatHighlight, vatUnderline, vatStrikeOut, vatLine, vatArrow, vatSquare, vatCircle, vatInk ou vatStamp.
AnnotationAuthor: string	Autor das anotações e respostas que o usuário adiciona.
AnnotationColor: TAlphaColor	Cor das novas anotações. TAlphaColorRec.Null (padrão) usa a cor habitual de cada ferramenta.
AnnotationStampName: string	Nome do carimbo colocado por vatStamp. Padrão Approved.
ShowAnnotationPopups: Boolean	Abre um popup quando se toca ou se clica em uma anotação com conteúdo ou respostas. Padrão True.

Propriedade	Descrição
OnAnnotationClick(Sender; AAnnotation: TsgcPDFAnnotation; var AHandled: Boolean)	Uma anotação com conteúdo ou respostas foi tocada ou clicada. Defina AHandled como True para ignorar o popup.
OnAnnotationSelect(Sender; AAnnotation: TsgcPDFAnnotation)	A anotação selecionada mudou. AAnnotation é nil quando a seleção foi limpa.
OnAnnotationAdd(Sender; AAnnotation: TsgcPDFAnnotation)	O usuário adicionou uma anotação com uma ferramenta, ou uma resposta no popup.
OnAnnotationChange(Sender; AAnnotation: TsgcPDFAnnotation)	O usuário moveu, redimensionou ou editou uma anotação.
OnAnnotationDelete(Sender; AAnnotation: TsgcPDFAnnotation; var AAllow: Boolean)	Uma anotação está prestes a ser excluída com suas respostas. Defina AAllow como False para mantê-la.

Métodos Públicos

LoadFromFile, LoadFromStream, Close, GotoPage, NextPage, PrevPage, FirstPage, LastPage, ZoomIn, ZoomOut, ClearCache, SetScrollPosition, GetPageRect, DeviceToPage, PageToDevice, PageRotation e CachedPageCount. Camadas (conteúdo opcional): GetLayers, SetLayerVisible e LayerVisible, como em TsgcPDFViewerControl. Somente leitura: PageCount, ScrollX, ScrollY, ContentWidth, ContentHeight e FileName.
Anotações: AnnotationAt(X, Y: Single), SelectedAnnotation (leitura e gravação), SelectAnnotation, DeleteSelectedAnnotation (retorna True quando a anotação foi excluída), ShowAnnotationPopup, CloseAnnotationPopup, GotoAnnotation e AnnotationsChanged, que você chama depois de alterar Document.Annotations no código. Funcionam como em TsgcPDFViewerControl: DeleteSelectedAnnotation funciona apenas com AnnotationEditing em True, e GotoAnnotation rola a anotação para a área visível e, com AnnotationEditing em True, também a seleciona.

Toque, Mouse e Teclado

- Arraste com o dedo ou o mouse para rolar e deslocar.
- Belisque para dar zoom (o controle habilita o gesto interativo Zoom).
- A roda do mouse rola, Shift + roda rola para os lados, Ctrl + roda dá zoom.
- Page Up, Page Down, Home, End e as setas navegam.
- Com AnnotationEditing em False, toque em um link para segui-lo, e toque em uma anotação com conteúdo ou respostas para abrir o seu popup.
- Com AnnotationEditing em True, um toque seleciona uma anotação e um toque em um espaço vazio limpa a seleção. Um clique duplo abre o popup. Arrastar a anotação selecionada a move, arrastar uma das suas alças a redimensiona, e arrastar em qualquer outro lugar faz a panorâmica. Delete remove a anotação selecionada, e as setas a movem 1 ponto, ou 10 pontos com Shift.
- Esc durante o arraste de uma ferramenta, de um movimento ou de um redimensionamento cancela apenas esse arraste. Caso contrário, fecha o popup da anotação, limpa a seleção e cancela a ferramenta de anotação.

Exemplo

```
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;
```

Anotações

O visualizador FMX tem o mesmo suporte a anotações que TsgcPDFViewerControl, com as mesmas ferramentas e eventos. Tocar ou clicar em uma anotação que tem conteúdo ou respostas abre um popup com seu autor, data, conteúdo e respostas. Com AnnotationEditing em True o texto do popup pode ser editado e sua caixa Reply adiciona uma resposta escrita por AnnotationAuthor.

- Com AnnotationEditing em True, um toque ou clique seleciona uma anotação e um clique duplo abre o seu popup. Arrastar a anotação selecionada a move, suas alças a redimensionam e Delete a remove com suas respostas. As setas movem a anotação selecionada 1 ponto, ou 10 pontos com Shift. Cada alteração dispara OnAnnotationChange.
- Esc durante o arraste de uma ferramenta, de um movimento ou de um redimensionamento cancela apenas esse arraste. Caso contrário, fecha o popup, limpa a seleção e cancela a ferramenta atual.
- AnnotationTool coloca uma anotação com um toque ou um arraste e depois volta para vatNone. Com AnnotationEditing em True, a nova anotação é selecionada. Uma nova nota abre o seu popup, quando ShowAnnotationPopups é True, para que se possa digitar o texto. Um novo texto livre abre um editor no local. Ctrl+Enter, ou um toque fora do editor, adiciona a anotação com o texto digitado. Esc, ou um texto vazio, a descarta.
- O visualizador FMX não tem seleção de texto, portanto vatHighlight, vatUnderline e vatStrikeOut marcam o texto dentro do retângulo que você arrasta, um quadrilátero por linha. Quando o retângulo não contém texto, nada é adicionado e a ferramenta continua ativa.
- As posições estão em coordenadas do controle (Single): AnnotationAt(X, Y) e PageToDevice(APageIndex, APDFX, APDFY): TPointF.

Salvar continua sendo tarefa do aplicativo, como no visualizador VCL: chame `Document.SaveIncremental` ou `Document.SaveToFile` quando `Document.Annotations.Modified` for True. Depois de salvar, `Document.Annotations.ResetModified` limpa esse indicador, e o próximo salvamento ainda grava todas as alterações feitas desde que o documento foi carregado. Veja Anotações. Para mostrar as anotações como uma lista de comentários ao lado do visualizador, vincule a ele um TsgcPDFCommentsViewFMX.

Exemplo: Anotações

```
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;
```

Renderizando em um Bitmap FMX

TsgcPDFRendererFMX (unidade sgcPDF_Renderer_FMX) renderiza uma página em um TBitmap FMX, por exemplo para uma pré-visualização personalizada ou para compartilhar uma imagem de página. O texto em fontes que não são incorporadas e não estão disponíveis como arquivos TrueType é desenhado com o layout de texto FMX da 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;
```

A seleção de texto, a busca e o preenchimento de formulário estão disponíveis apenas no visualizador VCL.

Veja também

- TsgcPDFViewerControl
- TsgcPDFCommentsViewFMX
- macOS, iOS e Android
- Arquitetura do Renderizador
- Anotações

TsgcPDFCommentsViewFMX

TsgcPDFCommentsViewFMX (unidade sgcPDF_ViewerFMX) é a versão FireMonkey de TsgcPDFCommentsView. É uma TTreeView que lista os comentários do documento de um TsgcPDFViewerFMX. Todas as anotações são listadas, exceto campos de formulário, links e popups. Requer Delphi XE5 ou posterior e é instalada pelos pacotes FMX, na página de paleta **sgcPDF**.

Membro	Descrição
Viewer: TsgcPDF-ViewerFMX	O visualizador a seguir. A lista é reconstruída quando o visualizador carrega ou fecha um documento e quando suas anotações mudam.
GroupByPage: Boolean	Lista os comentários sob um item por página, com o texto Page N (count). Quando False, os comentários formam uma lista única na ordem das páginas. Padrão True.
ShowReplies: Boolean	Mostra as respostas de um comentário como seus itens filhos. Quando False, apenas os comentários de nível superior são listados. Padrão True.
RebuildComments	Recarrega a lista a partir do documento do visualizador. A lista já acompanha o visualizador, e depois de alterar Document.Annotations no código o método AnnotationsChanged do visualizador também a reconstrói.
SelectedAnnotation: TsgcPDFAnnotation	Somente leitura. A anotação do item selecionado, nil quando nenhum item ou um item de página está selecionado.

Cada item mostra `Author, date: contents`. O conteúdo é exibido em uma linha, e textos longos são encurtados. Quando o conteúdo está vazio, é mostrado o tipo da anotação, como Highlight. Uma anotação sem autor mostra (no author), e a data é omitida quando a anotação não tem nenhuma.

- Selecionar um item rola o visualizador até a anotação com GotoAnnotation. Com AnnotationEditing do visualizador em True, a anotação também é selecionada no visualizador.
- Um clique duplo abre o popup da anotação.
- A tecla Delete exclui a anotação selecionada e suas respostas enquanto AnnotationEditing do visualizador for True. OnAnnotationDelete do visualizador pode recusar a exclusão.
- Quando o usuário seleciona uma anotação no visualizador, o item dela é selecionado na lista.

As demais propriedades e eventos, como Align, Width e OnChange, são os da TTreeView do FMX. Os itens são criados a partir do documento e não são armazenados no formulário.

Exemplo: Painel de comentários

```
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;
```

Veja também

- TsgcPDFViewerFMX
- TsgcPDFCommentsView
- Anotações

Fontes OpenType

`TsgcPDFDocument.AddTrueTypeFont` (uma fonte instalada) e `AddTrueTypeFontFromFile` (um arquivo .ttf, .otf ou a primeira face de um arquivo .ttc) aceitam ambas as variantes de fontes OpenType:

Contornos de fonte	Incorporada como
TrueType (tabela glyf, arquivos .ttf e a maioria dos .otf)	Fonte Type0 com um descendente CIDFontType2 e um programa FontFile2 em subconjunto.
PostScript / CFF (tabela CFF, arquivos .otf que começam com OTTO)	Fonte Type0 com um descendente CIDFontType0 e um programa FontFile3 em subconjunto do subtipo CIDFontType0C.

Em ambos os casos, apenas os glifos usados no documento são incorporados, um CMap ToUnicode é gravado para que o texto possa ser copiado e pesquisado, e as larguras dos glifos são gravadas em /W. `TsgcTrueTypeFont.IsCFF` informa qual variante uma fonte carregada possui. O texto é desenhado com `TsgcPDFPage.DrawTextTT` ou `DrawTextTTColor`, que também aplicam os recursos de layout OpenType descritos em Scripts Complexos e Shaping.

Exemplo

```
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 é a forma de usar fontes em servidores, no Linux e em plataformas móveis, onde a fonte pode não estar instalada. `FindTrueTypeFont(AFamilyName, ABold, AItalic)` retorna o índice de uma fonte já adicionada, ou -1.

Veja também

- Scripts Complexos e Shaping
- Texto em Chinês, Japonês e Coreano
- Texto e Fontes

Scripts Complexos e Shaping

Texto desenhado com uma fonte TrueType ou OpenType incorporada (`DrawTextTT`, `DrawTextTTColor`, TsgcPDFCanvas e TsgcPDFHTMLRenderer com fontes TrueType, e aparências de campos de formulário) passa pelo shaper OpenType do sgcPDF (unidade sgcPDF_Shaping, classe TsgcPDFShaper). Ele é escrito em Pascal puro e funciona em todas as plataformas. Você passa o texto na ordem lógica, como é digitado, e o shaper produz os glifos na ordem visual:

- **Árabe**: formas de junção (isol, init, medi, fina), ligaduras obrigatórias (rlig) como lam alef, e posicionamento de marcas.
- **Hebraico** e outros textos da direita para a esquerda, com caracteres espelhados como colchetes.
- **Texto bidirecional**: o algoritmo bidirecional do Unicode (UAX #9) ordena trechos mistos da esquerda para a direita e da direita para a esquerda, números incluídos. A direção do parágrafo vem do primeiro caractere forte.
- **Ligaduras e formas contextuais**: ccmp, locl, liga, clig e calt da tabela GSUB.
- **Kerning e posicionamento de marcas**: kern, mark, mkmk e curs da tabela GPOS, ou a tabela kern legada quando a fonte não tem kerning GPOS.

Quando o texto contém trechos da direita para a esquerda, o texto lógico também é gravado como /ActualText, de modo que copiar e colar e a extração de texto retornam o texto na ordem em que foi digitado.

Exemplo: Árabe, Hebraico e 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;
```

Usando o Shaper Diretamente

`TsgcTrueTypeFont.ShapeText(const AText: WideString; ABaseLevel: Integer = -1): TsgcPDFShapedGlyphs` retorna os glifos posicionados (GlyphID, Cluster, Text, XAdvance, YAdvance, XOffset, YOffset e o Level bidi) na ordem visual, com avanços em unidades de fonte. ABaseLevel é 0 para esquerda para direita, 1 para direita para esquerda e -1 para detectar. `EncodeShapedText(AText, AFontSize)` retorna os operadores de content stream para um texto moldado. A unidade sgcPDF_Shaping também expõe os auxiliares bidi `sgcPDFBidiVisual`, `sgcPDFBidiLevels` e `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 Formulário

Valores de campos de formulário com caracteres fora da codificação de fonte padrão (cirílico, grego, CJK, árabe, hebraico) recebem aparências desenhadas com uma fonte TrueType incorporada e moldadas da mesma forma. Defina `TsgcPDFAcroForm.AppearanceFontFile` ou `AppearanceFontName` para escolher a fonte. Quando ambas estão vazias, é usada a primeira fonte instalada que cobrir os caracteres (Arial Unicode MS, Segoe UI, Arial, Yu Gothic, MS Gothic, Microsoft YaHei, Malgun Gothic, Noto Sans).

Limitações

- Scripts que precisam de um mecanismo de shaping dedicado, como os scripts índicos (devanágari, bengali, tâmil) e os scripts do sudeste asiático (tailandês, khmer), são desenhados apenas com os recursos genéricos, sem reordenação de sílabas.
- Caracteres explícitos de incorporação e substituição bidi (LRE, RLE, LRO, RLO, isolados) são ignorados.

Veja também

- Fontes OpenType
- Texto em Chinês, Japonês e Coreano
- Formulários

Texto em Chinês, Japonês e Coreano

Para escrever texto em chinês, japonês ou coreano, adicione uma fonte que contenha os caracteres com `AddTrueTypeFont` ou `AddTrueTypeFontFromFile` e desenhe com `DrawTextTT`. A fonte é incorporada como um subconjunto (apenas os glifos usados), de modo que a saída permanece pequena mesmo com fontes CJK grandes, e é exibida da mesma forma em qualquer computador. Coleções TrueType (.ttc) são aceitas: a primeira face da coleção é usada.

```
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;
```

Fontes típicas do Windows: MS Gothic, Yu Gothic e Meiryo (japonês), SimSun e Microsoft YaHei (chinês simplificado), MingLiU e Microsoft JhengHei (chinês tradicional), Batang, Gulim e Malgun Gothic (coreano). No Linux, use por exemplo as fontes Noto Sans CJK ou WenQuanYi com AddTrueTypeFontFromFile.

Lendo e Renderizando Documentos CJK

A extração de texto e os renderizadores suportam documentos CJK criados por outras ferramentas: fontes compostas com codificações Identity-H e Identity-V e os CMaps CJK predefinidos, fontes CID incorporadas (TrueType e CFF), e escrita vertical. Documentos que usam fontes CJK sem incorporá-las são renderizados com uma fonte instalada do sistema.

TsgcPDFCJKFontManager

A unidade sgcPDF_Fonts_CJK (somente Windows) mantém as classes antigas de baixo nível TsgcPDFCIDFont e TsgcPDFCJKFontManager e os auxiliares `sgcPDFIsCJKCodePoint` e `TsgcPDFCJKFontManager.ContainsCJK`. Essas classes não estão conectadas a TsgcPDFDocument. Use AddTrueTypeFont e DrawTextTT para código novo.

Veja também

- Fontes OpenType
- Scripts Complexos e Shaping

Extração de Texto: Posições de Caracteres

Além do texto simples e dos fragmentos, **TsgcPDFTextExtractor** (unidade sgcPDF_TextExtract) retorna cada caractere com sua posição, agrupado em palavras e linhas na ordem de leitura. É isso que o visualizador VCL usa para seleção de texto e realce de pesquisa, e você pode usá-lo para pesquisa de texto com coordenadas, extração de dados de layouts fixos ou redação de texto encontrado.

API de Caractere, Palavra e Linha

Método	Descrição
GetCharCount, GetChar(AIndex): TsgcPDFTextChar	Caracteres na ordem de leitura de GetText. TsgcPDFTextChar tem o quadrilátero do glifo (X1, Y1 a X4, Y4, seguindo a direção do texto), a caixa delimitadora (Left, Bottom, Right, Top), a origem (OriginX, OriginY), FontSize, LineIndex, WordIndex (-1 para espaço em branco) e Synthetic (um espaço inserido pelo extrator entre dois trechos de texto).
GetWordCount, GetWord(AIndex): TsgcPDFTextWord	Palavras como um intervalo de caracteres (FirstChar, CharCount), com LineIndex e caixa delimitadora.
GetLineCount, GetLine(AIndex): TsgcPDFTextLine	Linhas como um intervalo de caracteres com sua caixa delimitadora.
GetCharsText: WideString	Os caracteres unidos com CRLF entre as linhas. Mesmo texto que GetText.
CharAt(AX, AY: Double; ATolerance: Double = 0): Integer	Índice do caractere em um ponto do espaço do usuário, ou -1. ATolerance amplia as caixas dos caracteres.
IncludeAnnotations: Boolean	Também extrai o texto exibido pelos streams de aparência de anotações visíveis (valores de campos de formulário, texto livre e carimbos). Padrão False.

`sgcPDFCreatePageTextExtractor(AParser, APageIndex)` cria um extrator para uma página e executa Extract (você o libera; nil quando a página não existe). Todas as posições estão no espaço do usuário PDF: pontos, com origem no canto inferior esquerdo.

Exemplo

```
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;
```

Extração Aprimorada

- O texto dentro de form XObjects é extraído, com a matriz do formulário aplicada às posições.
- /ActualText do conteúdo marcado substitui os glifos que ele cobre, de modo que ligaduras, árabe moldado e marcas de hifenização são extraídos como o texto original.
- Fontes sem um CMap ToUnicode são mapeadas por meio de sua codificação, /Differences e nomes de glifos.

Veja também

- TsgcPDFTextExtractor
- Extração de Texto
- TsgcPDFViewerControl

Limites para Arquivos Não Confiáveis

O sgcPDF é frequentemente usado para abrir arquivos que vêm de fora: uploads, anexos de e-mail, documentos digitalizados. Todo parser e decodificador verifica sua entrada e gera uma exceção em dados corrompidos em vez de ler fora dos limites. As alocações são limitadas por estas variáveis globais, que você pode reduzir (ou aumentar) antes de carregar documentos:

Variável	Unidade	Padrão	Limite
sgcPDFMaxDecodedSize: Int64	sgcPDF_Filters	268435456 (256 MB)	Maior tamanho decodificado de um stream (Flate, LZW, RunLength, ASCII85, ASCIIHex, CCITT, JBIG2). Protege contra bombas de descompactação.
sgcPDFJPEGMaxPixels: Int64	sgcPDF_JPEG	67108864	Maior quadro JPEG (largura x altura).
sgcPDFJPEGMaxMemory: Int64	sgcPDF_JPEG	1073741824 (1 GB)	Memória de trabalho do decodificador JPEG para uma imagem.
sgcPDFJPXMaxPixels: Int64	sgcPDF_JPX	67108864	Maior imagem JPEG 2000 (largura x altura).
CS_CANVAS_MAX_RASTER (constante)	sgcPDF_Renderer_Canvas	67108864	Maior raster construído pelos renderizadores, em pixels.

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

Outras proteções estão incorporadas: limites de recursão para form XObjects aninhados, padrões, glifos Type 3, árvores de páginas e outlines, detecção de ciclos em referências de objetos, e caches limitados nos visualizadores. Uma imagem danificada é ignorada ou desenhada parcialmente, e o restante da página ainda é renderizado. Essas variáveis são globais ao processo. Defina-as uma vez na inicialização, não enquanto outras threads estão carregando documentos.

Veja também

- Tratamento de Erros
- Decodificadores de Imagem

Decodificadores de Imagem

O sgcPDF inclui decodificadores em Pascal puro para toda compactação de imagem usada em arquivos PDF. Os renderizadores, os visualizadores, TsgcPDFImageExtractor e o otimizador os utilizam, e você pode chamá-los diretamente. Eles funcionam em todas as plataformas e não precisam de DLL.

Formato	Unidade e função	Suportado
JPEG (DCTDecode)	sgcPDF_JPEG: sgcPDFDecodeJPEG, TsgcPDFJPEGDecoder	Huffman baseline e progressivo, 8 bits, 1, 3 ou 4 componentes, quaisquer fatores de amostragem, marcadores de reinício, transformações JFIF e Adobe (YCbCr, YCCK, CMYK). Codificação aritmética, JPEG sem perdas e de 12 bits geram EsgcPDFJPEGError.
JPEG 2000 (JPXDecode)	sgcPDF_JPX: sgcPDFDecodeJPX	Arquivos JP2 e codestreams brutos, blocos (tiles), todas as ordens de progressão, wavelets reversíveis 5/3 e irreversíveis 9/7, subamostragem de componentes, de 1 a 16 bits, paletas, canais alfa, sYCC.
CCITT Grupo 3 e Grupo 4 (CCITTFaxDecode)	sgcPDF_CCITT: sgcPDFCCITTFaxDecode, sgcPDFCCITTFaxDecodeParms	Grupo 3 unidimensional (MH), Grupo 3 bidimensional (MR) e Grupo 4 (MMR), EndOfLine, EncodedByteAlign, EndOfBlock, BlackIs1.
JBIG2 (JBIG2Decode)	sgcPDF_JBIG2: sgcPDFJBIG2Decode	Regiões genéricas (aritmética e MMR), dicionários de símbolos e regiões de texto com codificação aritmética, /JBIG2Globals. Segmentos de meio-tom, padrão, refinamento e símbolo e texto codificados com Huffman não são suportados e geram uma exceção.

Assinaturas

- `sgcPDFDecodeJPEG(const AData: TBytes; out AWidth, AHeight, AComponents: Integer): TBytes` e uma sobrecarga com `AColorTransform: Integer` (o valor PDF /ColorTransform, -1 quando ausente). Retorna amostras de 8 bits intercaladas.
- `sgcPDFDecodeJPX(const AData: TBytes; out AInfo: TsgcPDFJPXInfo; AKeep16: Boolean = False): TBytes`. AInfo fornece Width, Height, Components, ColorComponents, HasAlpha, BitsPerComponent, ColorSpace (sgcJPXColorGray, sgcJPXColorRGB, sgcJPXColorCMYK, sgcJPXColorICC) e o perfil ICC.
- `sgcPDFCCITTFaxDecode(const AData: TBytes; AK, AColumns, ARows: Integer; AEndOfLine, AEncodedByteAlign, AEndOfBlock, ABlackIs1: Boolean; ADamagedRowsBeforeError: Integer = 0): TBytes`. AK < 0 é Grupo 4, 0 é Grupo 3 MH, > 0 é Grupo 3 MR. Retorna linhas de 1 bit alinhadas a um byte.
- `sgcPDFJBIG2Decode(const AData, AGlobals: TBytes; out AWidth, AHeight: Integer): TBytes`. Retorna linhas de 1 bit com a convenção do PDF (0 é preto).

Exemplo

```
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 os decodificadores rejeitam dados corrompidos com uma exceção e limitam suas alocações, consulte Limites para Arquivos Não Confiáveis.

Veja também

- TsgcPDFImageExtractor
- Importação de TIFF
- Recursos de Renderização Suportados

Importação de TIFF

TsgcPDFImage e TsgcPDFDocument.AddImage carregam arquivos TIFF além de JPEG, PNG e BMP. Isso facilita transformar documentos digitalizados e arquivos de fax em PDF.

Arquivos TIFF Suportados

- Little e big endian (II e MM), strips e tiles.
- Compactação: nenhuma, PackBits, LZW, Deflate (com preditor horizontal), CCITT Grupo 3 e Grupo 4, e JPEG.
- Imagens bilevel, em tons de cinza, RGB, paleta e CMYK, de 1 a 16 bits por amostra (16 bits são reduzidos para 8).
- Arquivos com várias páginas: cada página (IFD) pode ser adicionada.

Páginas CCITT e JPEG de faixa única são incorporadas sem recodificação, como streams CCITTFaxDecode e DCTDecode, de modo que uma página de fax digitalizada mantém seu tamanho e qualidade originais. Outras páginas são compactadas com Flate.

API

Membro	Descrição
TsgcPDFDocument.AddImage(const AFileName: string; APageIndex: Integer): Integer	Adiciona a página APageIndex (baseada em zero) de um arquivo TIFF com várias páginas. Outros formatos aceitam apenas a página 0.
TsgcPDFImage.TIFFPageCount(const AData: TBytes): Integer	Função de classe: número de páginas de um arquivo TIFF.
TsgcPDFImage.LoadTIFF(const AData: TBytes; APageIndex: Integer = 0)	Carrega uma página em um TsgcPDFImage (depois adicione-a com TsgcPDFDocument.AddImageObject).
TsgcPDFImage.IsCCITT: Boolean	True quando a página é incorporada como dados CCITT sem recodificação.

Exemplo: TIFF com Várias Páginas para 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;
```

Veja também

- Imagens
- Decodificadores de Imagem

TsgcPDFImageExtractor

TsgcPDFImageExtractor (unidade sgcPDF_ImageExtract) lista as imagens de um PDF e as salva como arquivos PNG, JPEG ou BMP, ou retorna seus pixels. Encontra image XObjects, imagens inline e imagens dentro de form XObjects (e opcionalmente dentro de anotações), com a posição onde cada imagem é pintada. Está na página de paleta do sgcPDF.

Propriedades

Propriedade	Descrição
Document: TsgcPDFDocument	Um documento carregado.
Parser: TsgcPDFParser	Um parser explícito. Tem precedência sobre Document.
IncludeInline: Boolean	Lista imagens inline (BI, ID, EI). Padrão True.
IncludeAnnotations: Boolean	Lista imagens pintadas por streams de aparência de anotações. Padrão False.
IncludeUnused: Boolean	Lista imagens dos recursos da página que o conteúdo nunca pinta. Padrão True.
UniqueOnly: Boolean	Lista cada objeto de imagem uma vez. UseCount informa quantas vezes ele é pintado. Padrão False.
IncludeAlpha: Boolean	Adiciona a transparência /SMask ou /Mask como um canal alfa. Padrão True.
MaxFormDepth: Integer	Aninhamento máximo de form XObjects. Padrão 16.
FileNamePattern: string	Nome de arquivo usado por ExtractAll, sem extensão: %p número da página, %i número da imagem, %n número da imagem dentro da página, %o número do objeto, %r nome do recurso.
Count, Items[AIndex]: TsgcPDFImageItem	As imagens encontradas. A varredura ocorre sob demanda.
Errors: TStringList	Imagens que ExtractAll não conseguiu salvar.

Métodos

Método	Descrição
Scan, ScanPage(APageIndex), Clear	Lista as imagens de todas as páginas ou de uma página.
SaveToFile(AIndex; const AFileName: string)	Escolhe o formato pela extensão: .png, .jpg, .jpeg, .bmp. As extensões .jp2, .jpx, .j2k, .jb2, .bin e .raw salvam os dados do stream como armazenados.
SaveAsPNG, SaveAsJPEG, SaveAsBMP, SaveToStream(AIndex; AStream; AFormat)	Salva uma imagem. AFormat: pifAuto (imagens JPEG como JPEG, JPEG 2000 como armazenado, as demais como PNG), pifPNG, pifJPEG, pifBMP ou pifRaw.

Método	Descrição
ExtractAll(const AFolder: string; AFormat = pifAuto): Integer	Salva todas as imagens listadas e retorna o número de arquivos gravados.
GetPixels(AIndex; out AWidth, AHeight, AChannels: Integer): TBytes	Pixels de 8 bits decodificados, de cima para baixo: 1 Gray, 2 Gray e alfa, 3 RGB, 4 RGBA.
GetRawBytes, GetJPEGBytes, GetPNGBytes, GetBMPBytes, CanSaveAsJPEG	Dados de imagem em memória. A saída JPEG está disponível para imagens DCTDecode, que são salvas sem recompactação.

TsgcPDFImageItem descreve uma imagem: PageIndex, ObjectNumber, Name, Source (pisXObject ou pisInline), Width, Height, BitsPerComponent, ColorSpace, Filters, Matrix, Rect (a área pintada em pontos), Placed, InAnnotation, HasSMask, HasMask, IsImageMask, UseCount e Stream.

Exemplo: Listar e Salvar Imagens

```
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;
```

Exemplo: Pixels

```
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;
```

Os codificadores também estão disponíveis como funções: `sgcPDFEncodePNG` e `sgcPDFEncodeBMP(const APixels: TBytes; AWidth, AHeight, AChannels: Integer): TBytes`.

Veja também

- Decodificadores de Imagem
- Imagens

TsgcPDFCanvas

TsgcPDFCanvas (unidade sgcPDF_Canvas) desenha em uma página PDF com uma API modelada no TCanvas da VCL: objetos Pen, Brush e Font, MoveTo e LineTo, Rectangle, Ellipse, Arc, Pie, TextOut, TextRect e assim por diante. Tudo é gravado como gráficos vetoriais e texto real no conteúdo da página. É a forma mais fácil de portar código de impressão TCanvas existente para PDF, e é multiplataforma (sem VCL, sem GDI).

Coordenadas

A origem do canvas é o canto **superior esquerdo** da página, e Y cresce para baixo, como em um TCanvas. Escolha a unidade no construtor: `Create(APage: TsgcPDFPage; AUnit: TsgcPDFCanvasUnit = pcuPoints; ADPI: Double = 96)`, com pcuPoints, pcuPixels (em ADPI), pcuMillimeters ou pcuInches. Translate, Rotate, Scale, MultiplyTransform, SetTransform e ResetTransform alteram a transformação.

Principais Membros

Membro	Descrição
Pen	Color, Width (0 desenha a linha visível mais fina), Style (ppsSolid, ppsDash, ppsDot, ppsDashDot, ppsDashDotDot, ppsClear), Cap, Join, MiterLimit, GapFill e GapColor.
Brush	Color e Style (pbsSolid, pbsClear).
Font	Name, Size (pontos), Style (pfsBold, pfsItalic, pfsUnderline, pfsStrikeOut) e Color.
Opacity: Double	Alfa de preenchimento e traço, de 0 a 1.
TrueTypeFonts: Boolean	Fontes diferentes de Arial/Helvetica, Times e Courier são incorporadas como TrueType quando instaladas. False mapeia todas as fontes para as 14 fontes padrão.
MoveTo, LineTo, Rectangle, FillRect, FrameRect, RoundRect, Ellipse, Arc, ArcTo, Pie, Chord, Polygon, Polyline, PolylineTo, PolyBezier, PolyBezierTo	Linhas e formas, com as convenções de parâmetros do TCanvas.
BeginPath, EndPath, CloseFigure, FillPath, StrokePath, StrokeAndFillPath, ClipPath	Caminhos.
IntersectClipRect, ExcludeClipRect, ResetClip, SaveState, RestoreState	Recorte e estado.
TextOut, TextOutEx, TextRect, TextWidth, TextHeight, TextAscent, TextDescent	Texto, com alinhamento, quebra de linha, avanços de caractere e rotação (TextOutEx).
DrawImage(X, Y, W, H; AImageIndex | AData: TBytes | AStream | AImage: TsgcPDFImage)	Imagens.
EndDraw	Fecha o grupo de recorte aberto. Chame-o quando terminar de desenhar uma página.

As cores são valores TsgcPDFCanvasColor no layout TColor ($00BBGGRR). Construa-as com `sgcPDFCanvasRGB(R, G, B)`.

Exemplo

```
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;
```

Veja também

- EMF para PDF
- TsgcPDFVCLCanvasHelper
- HTML para PDF
- Gráficos

EMF para PDF

TsgcPDFEMFPlayer (unidade sgcPDF_EMF, somente Windows) reproduz metarquivos aprimorados do Windows (EMF) em um TsgcPDFCanvas. Linhas, formas, caminhos, texto e bitmaps do metarquivo se tornam gráficos vetoriais, texto real e imagens no PDF, de modo que gráficos e relatórios desenhados por outros componentes permanecem nítidos em qualquer zoom e o texto pode ser pesquisado.

API

Membro	Descrição
Create(ACanvas: TsgcPDFCanvas)	O canvas que recebe o desenho.
Play(AHandle: HENHMETAFILE; X, Y, W, H: Double)	Reproduz um metarquivo de modo que seu quadro preencha o retângulo (unidades do canvas).
PlayFromFile, PlayFromStream, PlayFromBytes	O mesmo, a partir de um arquivo .emf, um stream ou bytes.
GetFrameSize(AHandle; out AWidth, AHeight: Double)	Procedimento de classe: tamanho do quadro do metarquivo em pontos.
RecordCount, SkippedRecords	Número de registros reproduzidos e de registros não suportados.
sgcPDFEMFAddPage(ADocument; AHandle | AFileName): TsgcPDFPage	Adiciona uma página com o tamanho do quadro do metarquivo e desenha o metarquivo nela.

Exemplo

```
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;
```

Veja também

- TsgcPDFCanvas
- TsgcPDFVCLCanvasHelper

TsgcPDFVCLCanvasHelper

TsgcPDFVCLCanvasHelper (unidade sgcPDF_CanvasVCL) conecta gráficos VCL a um TsgcPDFCanvas. Somente Windows e VCL.

Membro	Descrição
Create(ACanvas: TsgcPDFCanvas)	O canvas de destino.
DrawMetafile(AMetafile: TMetafile; X, Y, W, H: Double)	Reproduz um TMetafile como gráficos vetoriais.
DrawGraphic(AGraphic: TGraphic; X, Y, W, H: Double)	Desenha qualquer TGraphic: metarquivos como gráficos vetoriais, bitmaps, imagens PNG e JPEG como imagens.
DrawControl(AControl: TWinControl; X, Y, W, H: Double)	Desenha um controle e seus filhos (por exemplo um gráfico ou uma grade) como gráficos vetoriais, por meio de um metarquivo aprimorado.
ControlToMetafile(AControl: TWinControl): TMetafile	Função de classe: grava um controle em um novo metarquivo (você o libera).

X, Y, W e H estão em unidades do canvas.

Exemplo

```
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;
```

Veja também

- TsgcPDFCanvas
- EMF para PDF

HTML para PDF

TsgcPDFHTMLRenderer (unidade sgcPDF_HTML) converte HTML com CSS em páginas PDF paginadas. Possui seu próprio mecanismo de layout escrito em Pascal puro, portanto não precisa de navegador e funciona em todas as plataformas. Use-o para faturas, relatórios, cartas e arquivos de e-mail gerados a partir de modelos HTML. Está na página de paleta do sgcPDF.

Métodos

Método	Descrição
HTMLToPDFFile(const AHTML, APDFFileName: string; const ABasePath: string = '')	Cria um arquivo PDF a partir de uma string HTML.
HTMLFileToPDFFile(const AHTMLFileName, APDFFileName: string)	Cria um arquivo PDF a partir de um arquivo HTML.
RenderHTML(const AHTML: string; ADocument: TsgcPDFDocument; const ABasePath: string = ''): Integer	Anexa as páginas a um documento e retorna o número de páginas adicionadas. RenderHTMLW recebe uma WideString, RenderHTMLBytes recebe bytes brutos (a codificação vem do BOM, de <meta charset> ou da detecção de UTF-8) e RenderHTMLFile recebe um arquivo.

ABasePath resolve caminhos relativos de imagens e folhas de estilo. Em versões do Delphi anteriores a 2009, uma AnsiString pode conter UTF-8 ou Windows-1252, o que é detectado.

Propriedades

Propriedade	Descrição
PageWidth, PageHeight: Double	Tamanho da página em pontos. O @page size do CSS o sobrepõe.
Margins: TsgcPDFHTMLMargins	Left, Top, Right e Bottom em pontos. O @page margin do CSS os sobrepõe.
DefaultFontName, DefaultFontSize	Fonte do texto sem font-family ou font-size.
TrueTypeFonts: Boolean	Incorpora famílias de fontes diferentes das 14 padrão como fontes TrueType. Necessário para texto fora do conjunto de caracteres Windows-1252.
HeaderText, FooterText: string	Texto centralizado nas margens superior e inferior. {page} e {pages} são substituídos pelo número da página e pelo total de páginas.
MaxPages: Integer	Para após esse número de páginas.
OnPageHeader, OnPageFooter(Sender; ACanvas: TsgcPDFCanvas; APageNumber, APageCount: Integer)	Desenhe seu próprio cabeçalho e rodapé com um TsgcPDFCanvas.
LastPageCount	Páginas adicionadas pela última renderização.

HTML Suportado

- Elementos de bloco: 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 e li (disc, circle, square, decimal, decimal-leading-zero, alpha minúsculo e maiúsculo, latin e roman), dl, dt e dd.
- Elementos inline: span, a, b, strong, i, em, u, ins, s, strike, del, small, big, sub, sup, code, kbd, samp, tt, cite, var, dfn, br.
- Tabelas: table, caption, thead, tbody, tfoot, tr, th e td, com colspan e rowspan. As linhas de cabeçalho são repetidas em todas as páginas.
- Imagens: img com arquivos JPEG, PNG, BMP e TIFF (caminhos locais relativos a ABasePath) e URIs data: em base64, com width e height.
- Links: a href se torna uma anotação de link.
- Folhas de estilo: blocos <style>, atributos style e <link rel="stylesheet"> para arquivos locais.

CSS Suportado

- Seletores: elemento, classe, id, combinadores descendente e filho, :first-child, com especificidade, herança e !important.
- Modelo de caixa: margin, padding, border (width, style solid, dashed e dotted, cor, por lado), width, max-width, height, background-color, border-collapse e border-spacing.
- Texto: color, font, font-family, font-size (incluindo palavras-chave, em, rem, %, px, pt, mm, cm, in), font-weight, font-style, line-height, letter-spacing, text-align (incluindo justify), text-indent, text-decoration, vertical-align, white-space.
- Listas e display: list-style, list-style-type, display (block, inline, list-item, table e partes de table, none).
- Paginação: page-break-before, page-break-after, page-break-inside, break-before, break-after, break-inside, e @page com size (A3, A4, A5, B5, letter, legal, landscape) e margin. Linhas, imagens e linhas de tabela nunca são divididas entre páginas.
- Cores como nomes, #rgb, #rrggbb, rgb() e rgba().

Não suportado: JavaScript, floats, posicionamento absolute e fixed, flexbox, grid, fontes web (@font-face), imagens carregadas via HTTP, SVG e controles de formulário. Propriedades não suportadas são ignoradas.

Exemplo: String HTML para 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;
```

Exemplo: Anexar HTML a um 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;
```

Veja também

- TsgcPDFCanvas
- Tabelas

Códigos de Barras 2D e GS1-128

TsgcPDFBarcodeGenerator (unidade sgcPDF_Barcodes) desenha símbolos 2D e códigos de barras GS1 como gráficos vetoriais, além dos códigos de barras lineares descritos em Códigos de barras. O parâmetro APage de cada método Draw aceita um TsgcPDFPage ou seu construtor de conteúdo (`Page.Content`). Qualquer outro objeto gera uma exceção.

Simbologia	Método	Opções
QR Code	DrawQRCode(APage; const AData: AnsiString; AX, AY, AModuleSize: Double; AErrorLevel: TsgcPDFQRErrorLevel; AMode: TsgcPDFQRMode; AVersion: Integer)	Versões 1 a 40 (0 escolhe a menor que couber; uma versão fixa gera exceção quando os dados não cabem), correção de erro qrecL, qrecM, qrecQ, qrecH, modos qrmAuto, qrmNumeric, qrmAlphanumeric, qrmByte. Sobrecargas mais curtas usam os padrões.
DataMatrix ECC200	DrawDataMatrix(APage; const AData: AnsiString; AX, AY, AModuleSize: Double; AShape: TsgcPDFDataMatrixShape = dmsSquare)	24 tamanhos quadrados e 6 retangulares, dmsSquare, dmsRectangle ou dmsAuto (menor símbolo de qualquer formato). Codificação ASCII e Base256.
PDF417	DrawPDF417(APage; const AData: AnsiString; AX, AY, AModuleWidth: Double; ASecurityLevel: Integer = -1; AColumns: Integer = 0)	Níveis de segurança de 0 a 8 (-1 automático), de 1 a 30 colunas de dados (0 automático), de 3 a 90 linhas. Compactação de texto, numérica e de bytes.
GS1-128	DrawBarcode(APage; bcGS1_128; const AData: AnsiString; AX, AY: Double [; const AOptions])	Identificadores de aplicação entre parênteses, por exemplo (01)09501101530003(17)261231. O FNC1 é inserido e o dígito verificador do GTIN é verificado.

DataMatrix e PDF417 também estão disponíveis por meio de DrawBarcode com bcDataMatrix e bcPDF417, usando os campos QRVersion, QRErrorLevel, QRMode, DataMatrixShape, PDF417SecurityLevel, PDF417Columns e PDF417RowHeight de TsgcPDFBarcodeOptions. Os codificadores `EncodeQRCodeMatrix`, `EncodeDataMatrix`, `EncodePDF417` e `EncodeGS1_128`, e `EncodeLinear` para qualquer simbologia linear, retornam a matriz de módulos ou as larguras das barras, caso você queira desenhar o símbolo por conta própria. TsgcPDFBarcodeSymbol faz esse trabalho para qualquer simbologia: ele adiciona a zona de silêncio, a linha de texto e a rotação, e é o símbolo exibido pelos controles de código de barras VCL e FireMonkey.

Exemplo

```
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;
```

Não implementado: codificação C40 e Text do DataMatrix, e segmentação em modo misto do QR Code. A saída ainda é válida, apenas menos compacta para alguns dados.

Veja também

- Códigos de Barras
- TsgcPDFCanvas
- TsgcPDFBarcodeSymbol
- TsgcPDFBarcodeControl

TsgcPDFBarcodeControl

TsgcPDFBarcodeControl (unit sgcPDF_Barcode_VCL) é um controle VCL que exibe um código de barras ou um símbolo 2D em um formulário. Ele hospeda um TsgcPDFBarcodeSymbol, que codifica o texto com os codificadores de TsgcPDFBarcodeGenerator, e desenha um retângulo preenchido por sequência de módulos escuros, de modo que cada módulo começa e termina em um pixel inteiro e o símbolo permanece nítido para os leitores. Ele oferece suporte às onze simbologias do sgcPDF: Code 128, Code 39, EAN-13, EAN-8, UPC-A, ITF, Codabar, GS1-128, QR Code, DataMatrix e PDF417.
O controle desenha apenas com o canvas da VCL, sem GDI+, e por isso funciona a partir do Delphi 7. Ele é instalado pelos pacotes VCL, na página de paleta **sgcPDF**. O mesmo símbolo pode ser salvo como PNG ou SVG, ou desenhado em uma página PDF com os mesmos módulos exibidos na tela.

Propriedades Publicadas

Propriedade	Descrição
Symbology: TsgcPDFBarcodeType	A simbologia: bcCode128 (padrão), bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcGS1_128, ou as simbologias 2D bcQRCode, bcDataMatrix e bcPDF417.
Text: string	Os dados a codificar. QR Code, DataMatrix e PDF417 codificam os bytes UTF-8 do texto. Uma simbologia linear aceita apenas o seu próprio conjunto de caracteres, e o GS1-128 recebe os identificadores de aplicação entre parênteses.
QuietZone: Integer	A margem clara em módulos. -1 (padrão) usa o padrão da simbologia: 10 para símbolos lineares, à esquerda e à direita, e 4 para QR Code, 1 para DataMatrix e 2 para PDF417, nos quatro lados.
BarHeight: Integer	A altura das barras de um símbolo linear em módulos. 0 (padrão) preenche a altura disponível.
ShowText: Boolean	Desenha a linha legível sob as barras de um símbolo linear. Padrão True. Símbolos 2D não têm linha de texto.
Rotation: TsgcPDFBarcodeRotation	br0 (padrão), br90, br180 ou br270: giros de um quarto de volta no sentido horário. A linha de texto gira junto com o símbolo.
Fit: TsgcPDFBarcodeFit	bfFitInteger (padrão): o maior número inteiro de pixels por módulo que cabe, centralizado. bfModuleSize: exatamente ModuleSize pixels por módulo, centralizado. bfStretch: o símbolo preenche o controle e os módulos podem diferir em um pixel.
ModuleSize: Integer	Pixels por módulo para bfModuleSize. Padrão 2. É dimensionado com o formulário em telas de alta densidade.
Color: TColor	O fundo e os módulos claros. Padrão clWhite.
BarColor: TColor	Os módulos escuros. Padrão clBlack.
Font, ParentFont	A fonte, o estilo e a cor da linha de texto. Sua altura acompanha o tamanho do módulo (8 módulos em uma faixa de 12 módulos sob as barras), portanto Font.Size é ignorado. Sempre é usada uma fonte TrueType.
Transparent: Boolean	Deixa o fundo sem pintar, de modo que o pai aparece entre as barras. Padrão False.

Propriedade	Descrição
QRErrorLevel: TsgcPDFQRErrorLevel	Correção de erros do QR Code: qrecL, qrecM (padrão), qrecQ ou qrecH.
QRMode: TsgcPDFQRMode	Modo de codificação do QR Code: qrmByte (padrão), qrmAuto, qrmNumeric ou qrmAlphanumeric.
QRVersion: Integer	0 (padrão) escolhe a menor versão do QR Code que cabe, de 1 a 40 fixa a versão.
DataMatrixShape: TsgcPDFDataMatrixShape	dmsSquare (padrão), dmsRectangle ou dmsAuto (o menor símbolo de qualquer forma).
PDF417SecurityLevel: Integer	Nível de segurança do PDF417 de 0 a 8, -1 (padrão) é automático.
PDF417Columns: Integer	Colunas de dados do PDF417 de 1 a 30, 0 (padrão) é automático.
PDF417RowHeight: Integer	A altura de uma linha do PDF417 em módulos. Padrão 3.
Align, Anchors, Visible, Hint	Propriedades VCL padrão.

Eventos

Evento	Descrição
OnError: TsgcPDFBarcodeErrorEvent	Chamado com Sender e aError (Symbol.LastError) quando o texto não pode ser codificado, uma vez a cada alteração do símbolo. O controle então desenha uma moldura fina hachurada. O desenho nunca gera exceção.
OnClick: TNotifyEvent	O controle foi clicado.

Métodos Públicos

Método	Descrição
IsValid: Boolean	True quando o texto pode ser codificado. Symbol.LastError informa o motivo quando não pode.
PaintTo(aCanvas: TCanvas; const aRect: TRect)	Desenha o símbolo em aRect de qualquer canvas, por exemplo um bitmap ou o canvas de uma impressora, do modo como o controle se desenha: o fundo (a menos que Transparent esteja ativo), os módulos e a linha de texto, ou a moldura hachurada quando o texto é inválido.
SaveToPNG(const aFileName: string; aModuleSize: Integer), SaveToPNG(aStream: TStream; aModuleSize: Integer)	Salva o símbolo como imagem PNG com aModuleSize pixels por módulo, BarColor sobre Color. A linha de texto usa a fonte de Font e BarColor.
SaveToSVG(const aFileName: string; aModuleSize: Integer), SaveToSVG(aStream: TStream; aModuleSize: Integer)	Salva o símbolo como SVG com aModuleSize unidades por módulo, com as mesmas cores e a mesma fonte de SaveToPNG.

Método	Descrição
Symbol: TsgcPDFBarcodeSymbol	O símbolo hospedado, somente leitura. O controle é o seu dono e usa o seu OnChange. Use-o para desenhar o símbolo em uma página PDF (DrawToPDF) ou para ler Runs, ModuleWidth e ModuleHeight.

SaveToPNG e SaveToSVG geram uma exceção com Symbol.LastError quando o texto é inválido, antes de qualquer arquivo ser criado.

Exemplo

```
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;
```

Adicionando o controle a um PDF

TsgcPDFBarcodeSymbol.DrawToPDF desenha o símbolo do controle em uma página PDF como retângulos vetoriais, com os mesmos módulos, a mesma rotação, a mesma zona de silêncio e a mesma linha de texto da tela. DrawToPDF recebe as cores como $RRGGBB, enquanto um TColor da VCL é $BBGGRR, portanto troque antes os bytes de vermelho e 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;
```

Veja também

- TsgcPDFBarcodeFMX
- TsgcPDFBarcodeSymbol
- Códigos de Barras
- Códigos de Barras 2D e GS1-128

TsgcPDFBarcodeFMX

TsgcPDFBarcodeFMX (unit sgcPDF_Barcode_FMX) é a versão FireMonkey de TsgcPDFBarcodeControl. Ele hospeda o mesmo TsgcPDFBarcodeSymbol e desenha um retângulo por sequência de módulos escuros. O layout é calculado em pixels do dispositivo, com a escala do canvas, de modo que cada módulo começa e termina em um pixel inteiro do dispositivo e permanece nítido em telas de alta densidade. O controle limita o seu desenho aos seus limites.
Ele requer o Delphi XE5 ou posterior (em compiladores mais antigos a unit fica vazia) e é instalado pelos pacotes FMX, na página de paleta **sgcPDF**.

Propriedades Publicadas

Propriedade	Descrição
Symbology: TsgcPDFBarcodeType	A simbologia: bcCode128 (padrão), bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcGS1_128, ou as simbologias 2D bcQRCode, bcDataMatrix e bcPDF417.
Text: string	Os dados a codificar. QR Code, DataMatrix e PDF417 codificam os bytes UTF-8 do texto. Uma simbologia linear aceita apenas o seu próprio conjunto de caracteres, e o GS1-128 recebe os identificadores de aplicação entre parênteses.
QuietZone: Integer	A margem clara em módulos. -1 (padrão) usa o padrão da simbologia: 10 para símbolos lineares, à esquerda e à direita, e 4 para QR Code, 1 para DataMatrix e 2 para PDF417, nos quatro lados.
BarHeight: Integer	A altura das barras de um símbolo linear em módulos. 0 (padrão) preenche a altura disponível.
ShowText: Boolean	Desenha a linha legível sob as barras de um símbolo linear. Padrão True. Símbolos 2D não têm linha de texto.
Rotation: TsgcPDFBarcodeRotation	br0 (padrão), br90, br180 ou br270: giros de um quarto de volta no sentido horário. A linha de texto gira junto com o símbolo.
Fit: TsgcPDFBarcodeFit	bfFitInteger (padrão): o maior número inteiro de pixels do dispositivo por módulo que cabe, centralizado. bfModuleSize: ModuleSize unidades lógicas por módulo, arredondadas para pixels inteiros do dispositivo, centralizado. bfStretch: o símbolo preenche o controle e os módulos podem diferir em um pixel do dispositivo.
ModuleSize: Integer	Unidades lógicas por módulo para bfModuleSize, como qualquer outro tamanho do FireMonkey. Padrão 2.
Color: TAlphaColor	O fundo e os módulos claros. Padrão branco.
BarColor: TAlphaColor	Os módulos escuros. Padrão preto.
TextSettings: TsgcPDFBarcodeTextSettingsFMX	Font (família e estilo) e FontColor da linha de texto. Sua altura acompanha o tamanho do módulo, portanto Font.Size é ignorado.
Transparent: Boolean	Deixa o fundo sem pintar. Padrão False.
QRErrorLevel: TsgcPDFQRErrorLevel	Correção de erros do QR Code: qrecL, qrecM (padrão), qrecQ ou qrecH.

Propriedade	Descrição
QRMode: TsgcPDFQRMode	Modo de codificação do QR Code: qrmByte (padrão), qrmAuto, qrmNumeric ou qrmAlphanumeric.
QRVersion: Integer	0 (padrão) escolhe a menor versão do QR Code que cabe, de 1 a 40 fixa a versão.
DataMatrixShape: TsgcPDFDataMatrixShape	dmsSquare (padrão), dmsRectangle ou dmsAuto (o menor símbolo de qualquer forma).
PDF417SecurityLevel: Integer	Nível de segurança do PDF417 de 0 a 8, -1 (padrão) é automático.
PDF417Columns: Integer	Colunas de dados do PDF417 de 1 a 30, 0 (padrão) é automático.
PDF417RowHeight: Integer	A altura de uma linha do PDF417 em módulos. Padrão 3.
Align, Anchors, Margins, Position, Width, Height, Size, Visible, Hint, ShowHint	Propriedades FireMonkey padrão. Size é publicada a partir do Delphi XE7, e Hint e ShowHint a partir do Delphi XE6.

Eventos

Evento	Descrição
OnError: TsgcPDFBarcodeErrorEvent	Chamado com Sender e aError (Symbol.LastError) quando o texto não pode ser codificado, uma vez a cada alteração do símbolo. O controle então desenha uma moldura fina hachurada. O desenho nunca gera exceção.
OnClick: TNotifyEvent	O controle foi clicado ou tocado.

Métodos Públicos

Método	Descrição
IsValid: Boolean	True quando o texto pode ser codificado. Symbol.LastError informa o motivo quando não pode.
SaveToPNG(const aFileName: string; aModuleSize: Integer), SaveToPNG(aStream: TStream; aModuleSize: Integer)	Salva o símbolo como imagem PNG com aModuleSize pixels por módulo, BarColor sobre Color. A linha de texto usa a família de TextSettings.Font.
SaveToSVG(const aFileName: string; aModuleSize: Integer), SaveToSVG(aStream: TStream; aModuleSize: Integer)	Salva o símbolo como SVG com aModuleSize unidades por módulo, com as mesmas cores e a mesma fonte de SaveToPNG.
Symbol: TsgcPDFBarcodeSymbol	O símbolo hospedado, somente leitura. O controle é o seu dono e usa o seu OnChange.

SaveToPNG e SaveToSVG geram uma exceção com Symbol.LastError quando o texto é inválido, antes de qualquer arquivo ser criado.

Cores

Um TAlphaColor é $AARRGGBB, e os métodos de TsgcPDFBarcodeSymbol recebem $RRGGBB e ignoram o byte mais alto. Assim, Color e BarColor são passados como estão a DrawToPDF, SaveToSVG e SaveToPNG do símbolo, sem troca de bytes.

Exemplo

```
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;
```

Veja também

- TsgcPDFBarcodeControl
- TsgcPDFBarcodeSymbol
- macOS, iOS e Android

TsgcPDFBarcodeSymbol

TsgcPDFBarcodeSymbol (unit sgcPDF_Barcode_Symbol) é o código de barras independente de framework por trás de TsgcPDFBarcodeControl e TsgcPDFBarcodeFMX. Ele codifica um código de barras com os codificadores de TsgcPDFBarcodeGenerator, o organiza em uma grade de módulos e o grava como texto SVG, como imagem PNG ou como retângulos vetoriais em uma página PDF. Ele não usa VCL, FMX nem GDI, portanto faz parte do pacote core e também funciona em serviços, aplicativos de console e no Linux de 64 bits.

A grade de módulos

- A grade contém todo o símbolo após a rotação, incluindo a zona de silêncio e, para uma simbologia linear com ShowText, a faixa da linha legível.
- A faixa de texto tem 12 módulos de altura e fica sob as barras antes da rotação. O texto tem 8 módulos de altura, sua linha de base fica a 10 módulos da borda da faixa junto às barras e ele é centralizado ao longo da faixa. Caracteres de controle aparecem como espaços.
- Um símbolo linear com BarHeight 0 conta 50 módulos de barras em ModuleHeight e na saída SVG e PNG. Os controles e DrawToPDF fazem as suas barras preencherem a altura disponível.
- Rotation gira o símbolo no sentido horário em quartos de volta.
- As cores são $RRGGBB e o byte mais alto é ignorado, portanto um TAlphaColor do FireMonkey é passado como está, enquanto os bytes de vermelho e azul de um TColor da VCL precisam ser trocados.

Propriedades Publicadas

Propriedade	Descrição
Symbology: TsgcPDFBarcodeType	A simbologia: bcCode128 (padrão), bcCode39, bcEAN13, bcEAN8, bcUPCA, bcITF, bcCodabar, bcGS1_128, ou as simbologias 2D bcQRCode, bcDataMatrix e bcPDF417.
Text: string	Os dados a codificar. QR Code, DataMatrix e PDF417 codificam os bytes UTF-8 do texto. Uma simbologia linear aceita apenas o seu próprio conjunto de caracteres, e o GS1-128 recebe os identificadores de aplicação entre parênteses.
QuietZone: Integer	A margem clara em módulos. -1 (padrão) usa o padrão da simbologia: 10 para símbolos lineares, à esquerda e à direita, e 4 para QR Code, 1 para DataMatrix e 2 para PDF417, nos quatro lados.
BarHeight: Integer	A altura das barras de um símbolo linear em módulos. 0 (padrão) preenche a altura disponível.
ShowText: Boolean	Desenha a linha legível sob as barras de um símbolo linear. Padrão True. Símbolos 2D não têm linha de texto.
Rotation: TsgcPDFBarcodeRotation	br0 (padrão), br90, br180 ou br270: giros de um quarto de volta no sentido horário. A linha de texto gira junto com o símbolo.
QRErrorLevel: TsgcPDFQRErrorLevel	Correção de erros do QR Code: qrecL, qrecM (padrão), qrecQ ou qrecH.
QRMode: TsgcPDFQRMode	Modo de codificação do QR Code: qrmByte (padrão), qrmAuto, qrmNumeric ou qrmAlphanumeric.
QRVersion: Integer	0 (padrão) escolhe a menor versão do QR Code que cabe, de 1 a 40 fixa a versão.

Propriedade	Descrição
DataMatrixShape: TsgcPDFDataMatrixShape	dmsSquare (padrão), dmsRectangle ou dmsAuto (o menor símbolo de qualquer forma).
PDF417SecurityLevel: Integer	Nível de segurança do PDF417 de 0 a 8, -1 (padrão) é automático.
PDF417Columns: Integer	Colunas de dados do PDF417 de 1 a 30, 0 (padrão) é automático.
PDF417RowHeight: Integer	A altura de uma linha do PDF417 em módulos. Padrão 3.
OnChange: TNotifyEvent	Uma propriedade mudou. Os controles de código de barras usam o OnChange do seu Symbol para se redesenhar, portanto não o substitua ali.

Métodos e propriedades públicos

Membro	Descrição
Encode: Boolean	Codifica o símbolo quando uma propriedade mudou desde a última chamada e retorna True quando o texto é válido. Os outros métodos o chamam quando necessário.
LastError: string	O motivo pelo qual o texto não pode ser codificado.
ModuleWidth, ModuleHeight: Integer	O tamanho da grade girada em módulos, incluindo a zona de silêncio e a faixa de texto. 0 quando o texto é inválido.
ModuleAt(aCol, aRow: Integer): Boolean	True quando o módulo em aCol, aRow da grade girada é escuro.
Runs: TsgcPDFBarcodeRuns	Os módulos escuros como retângulos TsgcPDFBarcodeRun (Col, Row, Cols, Rows), mesclados por linha e depois entre linhas, ordenados por linha e coluna. Vazio quando o texto é inválido.
TextBounds(out aCol, aRow, aCols, aRows: Integer): Boolean	A faixa da linha de texto na grade girada. False quando o símbolo não desenha texto: simbologias 2D, ShowText desativado ou texto inválido.
FitModuleSize(aWidth, aHeight: Integer): Integer	O maior tamanho inteiro de módulo com o qual o símbolo cabe em aWidth por aHeight pixels, 0 quando um pixel por módulo não cabe. Para um símbolo linear com BarHeight 0, apenas a faixa de texto e um módulo de barras contam no sentido transversal às barras, porque as barras preenchem a altura.
LayoutAxes(aFit: TsgcPDFBarcodeFit; aModuleSize, aLeft, aTop, aWidth, aHeight: Integer; out aX, aY: TsgcPDFBarcodeAxis)	O layout completo em pixels da grade em uma caixa, aquele com o qual os controles desenham. Consulte Desenho personalizado.
SaveToSVG(const aFileName: string; aModuleSize: Integer; aFore, aBack: Cardinal; const aFontName: string), SaveToSVG(aStream: TStream; ...)	Grava o símbolo como SVG com aModuleSize unidades por módulo, aFore sobre aBack. A linha de texto usa a família de fontes aFontName.
SaveToPNG(const aFileName: string; aModuleSize: Integer; aFore, aBack: Cardinal; const aFontName: string), SaveToPNG(aStream: TStream; ...)	Grava o símbolo como imagem PNG com aModuleSize pixels por módulo, por meio do canvas raster em Pascal puro. A linha de texto usa a fonte TrueType do sistema operacional aFontName, ou uma

Membro	Descrição
	fonte sem serifa quando ela não está instalada. Sem nenhuma fonte utilizável, o PNG não tem linha de texto.
DrawToPDF(aPage: TObject; aX, aY, aWidth, aHeight: Double; aFore, aBack: Cardinal; aFont: TsgcPDFStandardFont = sfHelvetica)	Desenha o símbolo em uma página PDF. Consulte Desenhando em uma página PDF.
Assign(Source: TPersistent)	Copia as propriedades publicadas de outro símbolo.

SaveToSVG e SaveToPNG geram uma exceção quando aModuleSize é menor que 1, e com LastError quando o texto é inválido, antes de qualquer arquivo ser criado.

Desenhando em uma página PDF

DrawToPDF desenha o símbolo como gráficos vetoriais, de modo que uma página PDF mostra exatamente o que um controle de código de barras mostra. Ele substitui o mapeamento manual das propriedades de um controle para TsgcPDFBarcodeOptions: a rotação, a zona de silêncio, a altura das barras e a linha de texto vêm do símbolo. TsgcPDFBarcodeGenerator.DrawBarcode continua sendo a forma mais curta de desenhar um código de barras diretamente a partir de dados.

Parâmetro	Descrição
aPage: TObject	Um TsgcPDFPage ou o seu construtor de conteúdo, `TsgcPDFPage.Content`, como em DrawBarcode. Qualquer outro objeto gera uma exceção.
aX, aY: Double	O canto inferior esquerdo da caixa em pontos. O eixo y do PDF aponta para cima.
aWidth, aHeight: Double	O tamanho da caixa em pontos. Uma caixa vazia gera uma exceção.
aFore: Cardinal	A cor das barras e da linha de texto, $RRGGBB.
aBack: Cardinal	A cor pintada sob todo o símbolo, incluindo a zona de silêncio, $RRGGBB.
aFont: TsgcPDFStandardFont	A fonte PDF padrão da linha de texto. Padrão sfHelvetica.

- Tamanho: o símbolo usa o maior tamanho de módulo que cabe na caixa, não apenas em pontos inteiros, e fica centralizado nela. Um símbolo linear com BarHeight 0 preenche a caixa no sentido transversal às barras, como FitModuleSize e LayoutAxes.
- Saída: o retângulo aBack, depois um retângulo preenchido por entrada de Runs em aFore, de modo que cada borda de módulo fica exatamente sobre a grade de módulos, depois a linha de texto, tudo dentro de um estado gráfico salvo.
- Texto: a linha de texto é desenhada em aFont com tamanho de 8 módulos, sua linha de base a 10 módulos da borda da faixa junto às barras, centralizada na faixa, e gira no sentido horário junto com o símbolo.
- Erros: um texto inválido gera uma exceção com LastError e não desenha nada.

Exemplo

Uma etiqueta GS1-128 desenhada duas vezes em uma página PDF, a segunda vez girada um quarto de volta, e salva como SVG e PNG, sem formulário e sem controle:

```
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;
```

Desenho personalizado

LayoutAxes calcula o layout completo em pixels com o qual os controles desenham, para uma caixa de aWidth por aHeight pixels cujo canto superior esquerdo é aLeft, aTop. aFit define o tamanho do módulo (aModuleSize pixels com bfModuleSize), o símbolo é centralizado e um símbolo linear com BarHeight 0 preenche a caixa no sentido transversal às barras. Ela retorna um TsgcPDFBarcodeAxis por eixo, e `sgcPDFBarcodeAxisPos(aAxis, aIndex)` retorna o pixel da linha de grade aIndex (de 0 a Count). Sequências vizinhas compartilham as suas bordas, portanto não aparecem lacunas entre os 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 e constantes

Nome	Descrição
TsgcPDFBarcodeRotation	br0, br90, br180, br270: giros de um quarto de volta no sentido horário.
TsgcPDFBarcodeFit	bfFitInteger, bfModuleSize, bfStretch: como os controles e LayoutAxes dimensionam os módulos.
TsgcPDFBarcodeRun, TsgcPDFBarcodeRuns	Um retângulo de módulos escuros na grade girada (Col, Row, Cols, Rows) e o array que Runs retorna.
TsgcPDFBarcodeAxis	Como as linhas de grade de um eixo correspondem a pixels inteiros: Origin, Module (pixels por módulo) e Count (módulos); com Module 0 (bfStretch), Span pixels são distribuídos entre Count módulos. FillStart, FillCount e FillPixels descrevem a faixa que preenche a caixa, as barras de um símbolo linear com BarHeight 0.
sgcPDFBarcodeAxisPos(const aAxis: TsgcPDF-BarcodeAxis; aIndex: Integer): Integer	O pixel da linha de grade aIndex em um eixo de Layou-tAxes.
TsgcPDFBarcodeErrorEvent	`procedure(Sender: TObject; const aError: string) of object` , o evento OnError dos dois con-troles de código de barras.
CS_SGC_PDF_BARCODE_TEXT_BAND, CS_SGC_PDF_BARCODE_TEXT_SIZE, CS_SGC_PDF_BARCODE_TEXT_BASELINE	A linha de texto em módulos: altura da faixa 12, tama-nho da fonte 8, linha de base 10.
CS_SGC_PDF_BARCODE_BAR_HEIGHT	50, a altura das barras em módulos de um símbolo line-ar com BarHeight 0 em ModuleHeight, SVG e PNG.

Veja também

- TsgcPDFBarcodeControl
- TsgcPDFBarcodeFMX
- Códigos de Barras
- Códigos de Barras 2D e GS1-128

Linux

O núcleo do sgcPDF compila e funciona em Linux 64 bits (Delphi 10.2 Tokyo e posterior com a plataforma Linux64). Isso cobre o trabalho típico de servidor: criar, editar, mesclar, preencher, criptografar e ler documentos PDF, e renderizar páginas em imagens, por exemplo em um serviço web ou uma tarefa em lote.

Disponível no Linux

- TsgcPDFDocument, TsgcPDFEditor e o parser e o gravador, incluindo salvamentos incrementais, object streams compactados e arquivos com mais de 2 GB.
- Texto, fontes TrueType e OpenType (carregadas de arquivos com AddTrueTypeFontFromFile, ou das pastas de fontes do sistema), shaping, tabelas, códigos de barras, anotações, marcadores, camadas, formulários, anexos, PDF marcado, saída e validação PDF/A.
- Criptografia por senha (RC4 e AES), redação, o otimizador.
- Extração de texto com posições de caracteres, TsgcPDFImageExtractor e todos os decodificadores de imagem.
- TsgcPDFRendererRaster: página para PNG ou BMP, e pixels da página.
- TsgcPDFCanvas e TsgcPDFHTMLRenderer.
- Assinaturas digitais por meio do evento OnExternalSign (por exemplo com o sgcSign ou seu próprio HSM), consulte Assinaturas Digitais.

Somente Windows

- TsgcPDFRendererGDI, TsgcPDFViewer, TsgcPDFExporter e os controles VCL TsgcPDFViewerControl, TsgcPDFThumbnailView e TsgcPDFOutlineView.
- TsgcPDFEMFPlayer e TsgcPDFVCLCanvasHelper.
- Assinatura e verificação com certificados de arquivos PFX ou do repositório de certificados do Windows, e criptografia por certificado (chave pública), que usam a CryptoAPI do Windows. No Linux essas chamadas geram uma exceção.
- As classes CJK de baixo nível de sgcPDF_Fonts_CJK (o próprio texto CJK funciona por meio de AddTrueTypeFontFromFile).

Fontes no Linux

`AddTrueTypeFont` procura fontes instaladas em /usr/share/fonts e /usr/local/share/fonts. O renderizador raster também varre ~/.fonts e ~/.local/share/fonts, e outras pastas adicionadas com `sgcPDFRasterAddFontDirectory`. Servidores mínimos e contêineres frequentemente não têm nenhuma fonte: instale um pacote como fonts-dejavu-core, fonts-liberation ou fonts-noto, ou distribua suas próprias fontes com o aplicativo e use `AddTrueTypeFontFromFile`. Documentos com fontes incorporadas são renderizados corretamente sem nenhuma fonte instalada.

Exemplo: Miniaturas no Lado do Servidor

O mesmo código do 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;
```

Veja também

- macOS, iOS e Android
- TsgcPDFRendererRaster
- Instalação

macOS, iOS e Android

A partir do Delphi XE5, o sgcPDF pode ser usado em aplicativos FireMonkey para macOS, iOS e Android, e para Windows e Linux (FMXLinux). As mesmas unidades não Windows disponíveis no Linux estão disponíveis, além do visualizador FireMonkey.

Recurso	Unidades	Observações
Criar, editar, ler e extrair	Unidades principais do sgcPDF	Como no Linux. Carregue fontes de arquivos distribuídos com o aplicativo (AddTrueTypeFontFromFile), pois as plataformas móveis não expõem suas fontes do sistema como arquivos de forma portátil.
Controle de visualização	sgcPDF_ViewerFMX	TsgcPDFViewerFMX: rolagem, zoom por pinça, ajustar à largura ou à página, links.
Imagens de página	sgcPDF_Renderer_FMX, sgcPDF_Renderer_Raster	TsgcPDFRendererFMX renderiza em um TBitmap FMX, TsgcPDFRendererRaster em PNG, BMP ou pixels.
Assinaturas	sgcPDF_Signatures	Somente por meio de OnExternalSign (sem CryptoAPI).

Instalação

Instale os pacotes FMX (sgcPDFFMXDxx e dclsgcPDFFMXDxx) além dos pacotes principais para obter TsgcPDFViewerFMX na paleta. Para plataformas móveis, a IDE compila as unidades da pasta Source no seu aplicativo: adicione a pasta Source do sgcPDF ao caminho de biblioteca de cada plataforma (Android de 32 e 64 bits, iOS Device 64 bits, macOS 64 bits e ARM 64 bits, Linux 64 bits).

Distribuindo Arquivos

Use o Deployment Manager para adicionar arquivos PDF e fontes ao aplicativo, e abra-os a partir da pasta de documentos, por exemplo com `TPath.Combine(TPath.GetDocumentsPath, 'sample.pdf')` (unidade System.IOUtils).

Veja também

- TsgcPDFViewerFMX
- Linux
- Arquitetura do Renderizador

C++Builder

O sgcPDF suporta C++Builder 2007 a C++Builder 13 com pacotes de tempo de execução e de tempo de design, na plataforma Win32 para todas as versões e no Win64 a partir do C++Builder XE6 (incluindo a plataforma Win64 Modern do C++Builder 13).

Pacotes

A pasta Packages contém, para cada versão do C++Builder, o pacote principal de tempo de execução **sgcPDFCxx.cbproj**, o pacote de tempo de execução VCL **sgcPDFVCLCxx.cbproj** e o pacote de tempo de design **dclsgcPDFCxx.cbproj**, por exemplo sgcPDFC13.cbproj, sgcPDFVCLC13.cbproj e dclsgcPDFC13.cbproj. Construa os dois pacotes de tempo de execução, depois construa e instale o pacote de tempo de design. A construção gera o cabeçalho .hpp de cada unidade e os arquivos de importação .lib ou .a.

Usando o sgcPDF a Partir do C++

- Inclua o cabeçalho de cada unidade que você usar, por exemplo sgcPDF_Document.hpp, sgcPDF_Fonts.hpp e sgcPDF_Renderer_Raster.hpp.
- As classes e funções têm os mesmos nomes que no Delphi, no namespace de sua unidade (por exemplo `Sgcpdf_document::TsgcPDFDocument`). A maioria dos cabeçalhos também é trazida para o namespace global pelas diretivas using geradas.
- Crie componentes com `new` e libere-os com `delete`. Componentes que descendem de TComponent recebem um owner no construtor (passe NULL para nenhum).
- Os parâmetros WideString e AnsiString aceitam as classes de string do C++Builder de mesmo nome; TBytes é `System::DynamicArray<System::Byte>`.

Os cabeçalhos removem a definição das macros do Windows que colidem com nomes de métodos (DrawText, FindText, GetObject, GetCharWidth, CreateWindow, CertNameToStr), de modo que métodos como `TsgcPDFTextExtractor::FindText` podem ser chamados normalmente.

Veja também

- Instalação

Anexos

Um PDF pode carregar outros arquivos dentro de si: os dados de origem de um relatório, uma planilha, o XML de uma fatura. **TsgcPDFDocument.Attachments** (classe TsgcPDFAttachments, unidade sgcPDF_Attachments) adiciona, lista, extrai e exclui esses arquivos incorporados. Eles são gravados na árvore de nomes `/EmbeddedFiles` do catálogo e, com seu relacionamento, no array `/AF` do catálogo exigido pelo PDF/A-3.

Adicionando Arquivos

```
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	Descrição
AddFromBytes(AName; AData; AMimeType; ADescription; ARelationship)	Incorpora AData sob o nome AName. Apenas AName e AData são obrigatórios.
AddFromStream(AName; AStream; ...)	Incorpora o restante de AStream.
AddFromFile(AFileName; AMimeType; ADescription; ARelationship)	Incorpora um arquivo do disco. O anexo recebe o nome do arquivo.
Delete(AIndex), Clear, IndexOf(AName)	Remove anexos e localiza um pelo nome.

Os nomes devem ser exclusivos: adicionar um segundo arquivo com o mesmo nome gera uma exceção. Cada anexo armazena seu tamanho, as datas de criação e modificação e uma soma de verificação MD5, conforme recomenda a especificação.

Relationship (AFRelationship)

ARelationship, ou a propriedade `Relationship` de TsgcPDFAttachment, informa a um leitor como o arquivo se relaciona com o documento (PDF 2.0 e PDF/A-3):

Valor	Significado
afrSource	A fonte original do conteúdo do documento, por exemplo a planilha a partir da qual um relatório foi criado.
afrData	Dados usados para construir o documento, por exemplo os valores de um gráfico.
afrAlternative	Uma representação alternativa do conteúdo, por exemplo o XML de uma fatura eletrônica.
afrSupplement	Uma representação complementar, por exemplo uma cópia legível por máquina.
afrEncrypted-Payload	Um documento de conteúdo criptografado.
afrFormData	Dados associados aos campos de formulário do documento.
afrSchema	Uma definição de esquema de um objeto associado.
afrUnspecified	Nenhum relacionamento informado (padrão).

Lendo e Extraindo

Documentos carregados preenchem a lista Attachments automaticamente. Cada TsgcPDFAttachment expõe Name, Description, MimeType, Relationship, CreationDate, ModDate e Size, e retorna seus dados com `GetBytes`, `SaveToStream` ou `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;
```

Anexos e PDF/A

Nível	Regra
PDF/A-1b	Arquivos incorporados não são permitidos: adicionar um gera uma exceção.
PDF/A-2b	Somente arquivos que sejam eles próprios documentos PDF/A. Defina `Attachments.PDFACompliantFiles` como True para confirmar isso antes de adicioná-los; caso contrário, a adição gera uma exceção.

Nível	Regra
PDF/A-3b	Qualquer tipo de arquivo, cada um com seu AFRelationship no array /AF do catálogo.

Defina `ShowAttachmentsPanel` como True para gravar `/PageMode /UseAttachments`, para que o visualizador abra seu painel de anexos. Os anexos são preservados por documentos criptografados e por SaveIncremental.

Veja também

- Faturas Factur-X e ZUGFeRD
- PDF/A
- TsgcPDFDocument

Faturas Factur-X e ZUGFeRD

Factur-X (França) e ZUGFeRD 2 (Alemanha) são o mesmo formato híbrido de fatura eletrônica: um documento PDF/A-3 que as pessoas podem ler, com a fatura estruturada (UN/CEFACT Cross Industry Invoice XML) incorporada nele para o software de contabilidade. O XRechnung usa o mesmo contêiner com seu próprio nome de arquivo.
O sgcPDF constrói o contêiner PDF. O próprio XML da fatura é criado pelo seu aplicativo (ou pelo seu ERP) e passado como bytes para **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;
```

O Que o AddFacturX Faz

- Incorpora o XML como **factur-x.xml** (**xrechnung.xml** para fxpXRechnung, ou o AFileName que você passar) com o tipo MIME text/xml.
- Define o relacionamento exigido pelo perfil: **Data** para MINIMUM e BASIC WL, que não são faturas completas, e **Alternative** para BASIC, EN 16931, EXTENDED e XRECHNUNG.
- Adiciona o esquema de extensão XMP do Factur-X (fx:DocumentType, fx:DocumentFileName, fx:Version, fx:ConformanceLevel) aos metadados.
- Alterna um documento sem nível PDF/A para **PDF/A-3b**. Um documento já definido como PDF/A-1b ou PDF/A-2b gera uma exceção, já que o Factur-X exige PDF/A-3.
- Substitui um XML de fatura anterior: um documento contém uma fatura.

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

O parâmetro opcional AVersion define fx:Version (padrão 1.0). As regras do PDF/A se aplicam ao documento inteiro, portanto desenhe o texto com fontes TrueType incorporadas (AddTrueTypeFont e DrawTextTT).
Documentos carregados relatam a fatura em `Attachments.FacturX`, `FacturXProfile` e `FacturXFileName`.
O exemplo acima foi verificado com o veraPDF (PDF/A-3b: PASS) e a fatura foi extraída sem alterações pela biblioteca Python factur-x.

Veja também

- Anexos
- PDF/A

Salvamento Incremental e Saída Compacta

SaveIncremental

TsgcPDFDocument.SaveIncremental atualiza um PDF carregado anexando uma nova revisão: os bytes originais são gravados sem alteração, seguidos apenas pelos objetos novos e alterados, uma seção de referência cruzada do mesmo tipo da original (tabela ou stream) e um trailer com `/Prev`. As entradas `/ID` e `/Encrypt` são preservadas. Use-o para editar documentos assinados sem quebrar suas assinaturas, para adicionar anotações ou valores de formulário a arquivos grandes rapidamente, e para manter o histórico de um 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;
```

Há duas sobrecargas: `SaveIncremental(AStream: TStream)` grava o arquivo atualizado completo em um stream, e `SaveIncremental(AFileName: string)` grava um arquivo, que pode ser o mesmo arquivo do qual o documento foi carregado. Um documento sem alterações produz os bytes originais.
As edições de anotações são salvas da mesma maneira. Uma anotação alterada do arquivo é gravada novamente com o seu próprio número de objeto, e as anotações e respostas novas são adicionadas na nova revisão, de modo que o intervalo de bytes de cada assinatura existente permanece o mesmo. Um validador aceita a revisão quando a assinatura permite comentários: uma assinatura de certificação com permissão DocMDP P=3, ou apenas assinaturas de aprovação. Com P=1 ou P=2 ele informa uma alteração não permitida. Veja Anotações.

Limitações

SaveIncremental gera uma exceção, sendo necessário um salvamento completo (SaveToFile ou SaveToStream), quando:

- o documento não foi carregado de um PDF (um documento novo);
- outros documentos foram anexados com AppendDocument;
- novas camadas ou novos campos de formulário foram adicionados (preencher campos existentes não tem problema);
- a criptografia foi alterada;
- o documento foi redigido após ter sido aberto com uma senha (consulte TsgcPDFRedactor).

CompressObjects

Com **CompressObjects** definido como True, um salvamento completo grava os objetos que não são streams em object streams compactados e a referência cruzada como um cross-reference stream compactado (PDF 1.5). Documentos com muitos objetos pequenos (páginas, anotações, campos de formulário, árvores de estrutura) ficam de 39 a 57 por cento menores nos testes. A criptografia é suportada. A propriedade é ignorada para PDF/A-1, que não 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;
```

Veja também

- Arquivos Grandes
- Carimbos de Tempo e Validação de Longo Prazo
- Otimizando Arquivos PDF
- Anotações

Arquivos Grandes

O sgcPDF usa deslocamentos de arquivo de 64 bits em toda parte, portanto lê e grava arquivos maiores que 2 GB, e não precisa carregar um arquivo grande na memória para trabalhar com ele.

MemoryThreshold

TsgcPDFDocument.LoadFromFile carrega na memória arquivos de até **MemoryThreshold** bytes (padrão 64 MB). Um arquivo maior é aberto para leitura compartilhada e seus objetos são lidos sob demanda por meio de um cache de leitura, de modo que abrir um arquivo de 3 GB lê apenas sua referência cruzada e os objetos que você usa. Defina MemoryThreshold como 0 para ler todos os arquivos sob 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;
```

Comportamento

- `Parser.StreamBacked` é True quando o documento é lido sob demanda, e `Parser.Size` é o tamanho do arquivo como Int64.
- O arquivo permanece aberto até que outro documento seja carregado ou o documento seja liberado. Outros processos podem lê-lo, mas não excluí-lo ou sobrescrevê-lo nesse meio tempo.
- SaveIncremental funciona em um documento lido sob demanda e pode gravar no mesmo arquivo: os bytes originais são copiados em blocos, o arquivo é substituído e o parser reabre o novo arquivo.
- LoadFromStream e LoadFromBytes sempre carregam na memória. Para ler um stream sob demanda, use o parser diretamente: `TsgcPDFParser.LoadFromStream(AStream, AOwnsStream)`. O stream deve permanecer ativo e inalterado enquanto o parser é usado.
- Um processo de 32 bits não consegue armazenar 2 GB em um único array: carregar uma fonte assim na memória gera uma exceção que solicita a leitura sob demanda.

Veja também

- Salvamento Incremental e Saída Compacta
- Tratamento de Erros e Entradas Hostis

FDF e XFDF

Os arquivos FDF (ISO 32000, 12.7.8) e XFDF (ISO 19444-1, XML) contêm apenas os valores dos campos de formulário de um PDF. Use-os para enviar dados preenchidos a um servidor, pré-preencher um formulário a partir de um banco de dados ou mover valores entre duas versões de um formulário. TsgcPDFAcroForm (`TsgcPDFDocument.AcroForm`) exporta e importa ambos os formatos.

Exportando

```
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	Descrição
ExportFDF(AStream or AFileName; ASourceFile)	Grava um arquivo FDF com o valor de cada campo sob seu nome totalmente qualificado. ASourceFile (opcional) é gravado como /F.
ExportXFDF(AStream or AFileName; ASourceFile)	Grava um arquivo XFDF (UTF-8). ASourceFile (opcional) é gravado como <f href>.

Importando

```
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` definem o valor de cada campo correspondente pelo seu nome totalmente qualificado (por exemplo `address.city`) e retornam o número de valores definidos. Nomes que não correspondem a nenhum campo são contados em `ImportUnknownCount`. As aparências dos campos alterados são regeneradas quando o documento é salvo.
Os valores de texto são Unicode em ambos os formatos, caixas de seleção e botões de opção usam seu valor de exportação (por exemplo Off ou Yes), e caixas de lista de seleção múltipla gravam um valor por opção selecionada.

Veja também

- Valores e Aparências de Formulário
- Formulários Interativos

Valores e Aparências de Formulário

```
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;
```

Caixas de Lista de Seleção Múltipla

Uma caixa de lista com o flag MultiSelect (bit 22 de `Flags`, valor $200000) pode conter vários valores. `SetValues` os seleciona, `ValueCount` e `GetValues(AIndex)` os leem, e `Value` retorna o primeiro. Os valores são gravados como um array em /V e como os índices /I das opções selecionadas, e são preservados por FDF e XFDF.

Aparências Unicode

A aparência de um valor de campo com caracteres fora da codificação de fonte padrão (cirílico, grego, CJK, árabe, hebraico...) é desenhada com uma fonte TrueType ou OpenType incorporada, registrada como `/SgcU` nos recursos de aparência e no dicionário /DR do formulário, de modo que o valor seja exibido e impresso corretamente também por visualizadores que não reconstroem aparências.

Propriedade	Descrição
AppearanceFontFile: string	Arquivo de fonte (.ttf, .otf) usado para esses valores. Tem precedência sobre AppearanceFontName.
AppearanceFontName: WideString	Fonte instalada usada para esses valores.

Quando ambas estão vazias, é usada a primeira fonte instalada dentre Arial Unicode MS, Segoe UI, Arial, Yu Gothic, MS Gothic, Microsoft YaHei, Malgun Gothic e Noto Sans que cobrir os caracteres. No Linux ou em um servidor sem essas fontes, defina AppearanceFontFile.

NeedAppearances

`AcroForm.NeedAppearances` grava `/NeedAppearances true`, que pede ao visualizador para reconstruir as aparências dos campos ao abrir o arquivo. É True por padrão para formulários novos. Documentos carregados mantêm o valor do arquivo. O sgcPDF sempre grava suas próprias aparências, de modo que os campos também aparecem

corretamente em visualizadores que ignoram o flag; defina como False para arquivos PDF/A, já que os padrões PDF/A não permitem que seja true.

Veja também

- FDF e XFDF
- Formulários Interativos

Assinaturas PAdES

PAdES (ETSI EN 319 142-1) é o perfil europeu de assinaturas PDF usado para assinaturas eletrônicas avançadas e qualificadas sob o eIDAS. TsgcPDFSignatureHandler cria assinaturas PAdES baseline B-B com a CryptoAPI do Windows, aceita um CMS criado por qualquer assinante externo, e adiciona carimbos de tempo de documento e dados de validação para os níveis B-T, B-LT e B-LTA.

SubFilter

SubFilter (TsgcPDFSignatureSubFilter)	Gravado como	Uso
pssPKCS7Detached (padrão)	/adbe.pkcs7.detached	Assinaturas PDF clássicas, compatíveis com todo leitor.
pssCAdESDetached	/ ETSI.CAdES.detached	Assinaturas PAdES baseline. O CMS carrega o 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;
```

Assinantes Externos (OnExternalSign)

Quando a chave privada não está disponível para a CryptoAPI (um serviço de assinatura remoto, um HSM, um middleware de smart card, um cofre de chaves na nuvem), atribua **OnExternalSign**. Nenhum certificado precisa ser carregado: o sgcPDF prepara o dicionário de assinatura e os intervalos de bytes, e o evento retorna o 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;
```

Parâmetro	Descrição
AData	Os bytes cobertos por /ByteRange (o arquivo sem o placeholder /Contents).
ADigest	O digest de AData com HashAlgorithm, para serviços que assinam um hash.
ASignature	Retorne um CMS SignedData destacado, completo e codificado em DER. Deve caber em PlaceholderSize bytes (padrão 8192; aumente para cadeias de certificados longas ou carimbos de tempo incorporados).

Para assinaturas PAdES completas construídas com o sgcSign (qualquer provedor de chaves, carimbos de tempo de assinatura, OCSP), use a ponte TsgcPDFPAdESSigner.

Veja também

- Carimbos de Tempo e Validação de Longo Prazo
- Verificação de Assinatura
- Assinaturas Digitais
- TsgcPDFSignatureHandler

Carimbos de Tempo e Validação de Longo Prazo

Uma assinatura só pode ser validada enquanto seu certificado é válido e seu status de revogação pode ser verificado. A validação de longo prazo (LTV) mantém uma assinatura verificável depois disso, incorporando os dados de validação no documento e protegendo tudo com um carimbo de tempo. TsgcPDFSignatureHandler grava ambos como atualizações incrementais, de modo que as assinaturas anteriores permanecem válidas.

Carimbos de Tempo de Documento

TimestampDocument (e TimestampDocumentStream) anexa uma assinatura invisível do tipo `/DocTimeStamp` com SubFilter `/ETSI.RFC3161`. O token de carimbo de tempo vem do evento **OnTimestamp**: o sgcPDF passa o digest a ser carimbado, e seu código envia uma solicitação RFC 3161 a uma autoridade de carimbo de tempo (TSA) e retorna o TimeStampToken da resposta.

```
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 é usado apenas para carimbos de tempo de documento. Carimbos de tempo de assinatura dentro do CMS (PAdES B-T) são adicionados por TsgcPDFPAdESSigner.

Document Security Store (DSS)

O DSS (`/DSS` no catálogo) contém os certificados, respostas OCSP e CRLs que um validador precisa, além de uma entrada `/VRI` para cada assinatura do 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	Descrição
AddDSSCertificate(AData)	Adiciona um certificado DER (CA intermediária, CA raiz, certificado da TSA).
AddDSSOCSPResponse(AData)	Adiciona uma resposta OCSP DER.
AddDSSCRL(AData)	Adiciona uma lista de revogação de certificados DER.
ClearDSS	Limpa os dados coletados.
AddDSSDocument(AInputFile, AOutputFile), AddDSSStream(AInput, AOutput)	Grava o DSS como uma atualização incremental de um documento assinado.

Níveis

Nível	Como
B-B	Assine com SubFilter pssCAdESDetached.
B-T	B-B com um carimbo de tempo de assinatura, ou um carimbo de tempo de documento adicionado logo após a assinatura.
B-LT	Adicione o DSS com a cadeia e os dados de revogação.
B-LTA	Adicione um carimbo de tempo de documento após o DSS. Repita o DSS e o carimbo de tempo antes que o certificado do carimbo de tempo expire.

A saída dos exemplos desta página foi validada com o pyHanko: a assinatura CAdES está intacta, válida e confiável, as modificações após ela são reconhecidas apenas como atualizações LTA, e o carimbo de tempo do documento está intacto e confiável.

Veja também

- Assinaturas PAdES
- TsgcPDFPAdESSigner
- Verificação de Assinatura

Verificação de Assinatura

TsgcPDFSignatureVerifier verifica cada assinatura e carimbo de tempo de documento de um PDF. `GetSignature(AIndex)` retorna um TsgcPDFSignatureInfo com os detalhes de cada um.

```
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;
```

Propriedade	Descrição
FieldName	Nome do campo de assinatura.
SubFilter	Por exemplo adbe.pkcs7.detached, ETSI.CAdES.detached ou ETSI.RFC3161.
IsDocTimeStamp	True para um carimbo de tempo de documento (/Type /DocTimeStamp). Seu token é verificado em relação aos bytes cobertos.
SignDate	A entrada /M da assinatura, ou o genTime (UTC) do token de um carimbo de tempo de documento.
CertInfo	SubjectName, IssuerName, SerialNumber, NotBefore, NotAfter e Thumbprint do certificado do assinante (ou da TSA).
IsValid	A assinatura CMS corresponde aos intervalos de bytes assinados.
IsTrusted	O certificado do assinante encadeia com uma raiz confiável para o Windows. Verificado separadamente: uma assinatura válida de um assinante desconhecido é válida, mas não confiável.
IsModified	A assinatura não cobre o arquivo inteiro. Atualizações incrementais após a assinatura (preenchimento de formulário, anotações, DSS, carimbos de tempo) a definem.
Name, Reason, Location	Os valores do dicionário de assinatura.

VerifySignature(AIndex) retorna IsValid. IsSignatureTrusted(AIndex) e IsDocumentModified(AIndex) retornam os outros dois resultados. A confiança é avaliada com o repositório de certificados do Windows, portanto certificados

de teste são relatados como não confiáveis. Para validação de política PAdES em relação a uma lista de confiança específica, use um validador dedicado como o sgcSign.

Veja também

- Assinaturas PAdES
- TsgcPDFSignatureVerifier

TsgcPDFPAdESSigner (Ponte sgcSign)

TsgcPDFPAdESSigner (unidade **sgcPDF_Sign_sgcSign**) cria assinaturas PAdES baseline B-B, B-T, B-LT e B-LTA com o **sgcSign**, a biblioteca de assinatura digital da eSeGeCe. O sgcPDF cuida do lado do PDF (campo, aparência, atualizações incrementais, DSS, carimbo de tempo de documento) e o sgcSign cuida da criptografia: o CMS com os atributos que o PAdES exige, o provedor de chaves, e os clientes de TSA e OCSP.
Como a chave privada é acessada por meio de um provedor de chaves do sgcSign, todo repositório de chaves que o sgcSign suporta funciona: arquivos PFX e PEM, tokens PKCS#11 e smart cards, o repositório de certificados do Windows e HSMs na nuvem.

Requisitos

- Uma licença do sgcSign e sua pasta Source no caminho de busca do projeto, junto ao sgcPDF.
- sgcPDF_Sign_sgcSign.pas é uma unidade opcional: **não** faz parte dos pacotes do sgcPDF. Adicione-a ao seu projeto.

Exemplo

```
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 exemplo foi compilado com o Delphi 13 em relação às fontes do sgcPDF e do sgcSign e executado com uma TSA pública real: o pyHanko valida o resultado como uma assinatura ETSI.CAdES.detached intacta e confiável, com um carimbo de tempo de assinatura válido e um /DSS.

Membro	Descrição
KeyProvider: IsgcKeyProvider	O provedor de chaves do sgcSign com o certificado do assinante e a chave privada.
Level: TsgcPDFPAdESLevel	paBB, paBT, paBLT ou paBLTA. Padrão paBB.

Membro	Descrição
TSAURL: string / TSAClient: IsgcTSAClient	Autoridade de carimbo de tempo para B-T e superiores: uma URL (é usado um cliente TSA interno do sgcSign) ou seu próprio cliente.
OCSPClient: IsgcOCSPClient	Opcional. Quando atribuído, a resposta OCSP do certificado do assinante é obtida e armazenada para B-LT.
HashAlgorithm	shaSHA256 (padrão), shaSHA384 ou shaSHA512.
AddCertificate, AddCRL, AddOCSPResponse, ClearValidationData	Material de validação. Os certificados também são incorporados no CMS; CRLs e respostas OCSP vão para o /DSS.
Handler: TsgcPDFSignatureHandler	Campo de assinatura, aparência, motivo, local e tamanho do placeholder.
SignPDF(AInput, AOutput), SignPDFFile(AInputFile, AOutputFile)	Assina no nível selecionado.
ExtendToLTA(AInput, AOutput)	Adiciona o DSS e um carimbo de tempo de documento a um PDF já assinado.

A ponte não grava um atributo CMS signing-time, que o PAdES proíbe junto à entrada /M do dicionário de assinatura.

Veja também

- Assinaturas PAdES
- Carimbos de Tempo e Validação de Longo Prazo

Criptografia por Certificado

A criptografia por senha compartilha um segredo com todo leitor. A criptografia por certificado (chave pública), o manipulador de segurança `/Adobe.PubSec`, criptografa a chave do documento para o certificado X.509 de cada destinatário: somente os proprietários das chaves privadas correspondentes podem abrir o arquivo, nenhuma senha é trocada, e cada destinatário pode receber permissões diferentes.

Criptografando

```
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	Descrição
AddEncryptionRecipient(ACertificate: TBytes; APermissions)	Adiciona um destinatário a partir de um certificado DER ou PEM.
AddEncryptionRecipient(ACertificateFile: string; APermissions)	O mesmo, a partir de um arquivo .cer, .der ou .pem.
AddEncryptionRecipientFromStore(ACertificateId; APermissions; AStoreName)	Um certificado do repositório do Windows, pela impressão digital SHA-1 (hex) ou parte do titular. AStoreName tem como padrão MY.
SetCertificateEncryption(AMethod)	Ativa a criptografia para os destinatários adicionados. emAES_256 e emAES_128 gravam /SubFilter /adbe.pkcs7.s5 (crypt filters), emRC4_128 e emRC4_40 o /adbe.pkcs7.s4 legado.

APermissions é um conjunto de TsgcPDFPermission: ppPrint, ppModify, ppCopy, ppAnnotate, ppFillForms, ppExtract, ppAssemble e ppPrintHigh.

Abrindo

```
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;
```

Defina a chave privada antes de LoadFromFile ou LoadFromStream com `SetDecryptionCertificate(APFXFileName, APassword)`, `SetDecryptionCertificate(APFX: TBytes, APassword)` ou `SetDecryptionCertificateFromStore(AStoreName)`, que testa os certificados com chave privada do repositório do usuário atual. `Encryption.Permissions` então contém as permissões do destinatário que abriu o arquivo.

Plataforma

A criptografia e a descriptografia por certificado usam a CryptoAPI do Windows (crypt32.dll, parte do Windows), portanto estão disponíveis somente no Windows. O arquivo produzido no exemplo foi descriptografado de forma independente pelo pyHanko com o PFX do primeiro destinatário.

Notas de Segurança

- As chaves de arquivo, os vetores de inicialização AES e os salts vêm do CSPRNG do sistema operacional (CryptGenRandom no Windows, /dev/urandom no Linux, macOS, iOS e Android).
- emAES_256 grava a revisão 6 do manipulador de segurança do PDF 2.0 (ISO 32000-2). Arquivos com a revisão 5 obsoleta (Acrobat 9) ainda são abertos.
- A criptografia só pode ser alterada por um salvamento completo: SaveIncremental gera uma exceção quando ela foi alterada.

Veja também

- Criptografia
- TsgcPDFRedactor

TsgcPDFRedactor

TsgcPDFRedactor (unidade sgcPDF_Redaction) realiza redação real: o texto, os pixels de imagem, os gráficos vetoriais e as anotações dentro das áreas marcadas são removidos do arquivo, não apenas cobertos por uma caixa preta. O conteúdo redigido não pode ser selecionado, copiado, extraído ou recuperado a partir da saída.

```
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;
```

Marcação

Método	Descrição
AddArea(APageIndex; ARect: TsgcPDFRect)	Um retângulo em pontos, no espaço de usuário padrão da página (origem no canto inferior esquerdo, antes do /Rotate da página).
AddArea(APageIndex; ALeft, ABottom, ARight, ATop)	O mesmo com coordenadas.
AddQuad(APageIndex; AQuad: TsgcPDFRedactionQuad)	Qualquer quadrilátero convexo, para texto rotacionado. sgcPDFRedactionQuadFromRect converte um retângulo.
AddText(AText; ACaseSensitive; AWholeWord; APageIndex)	Toda ocorrência de AText em uma página, ou em todas as páginas quando APageIndex é -1. As correspondências são buscadas dentro de uma linha de texto, e sequências de espaço em branco correspondem a qualquer espaço em branco.
Clear	Remove todas as marcas.

Aplicando

Método	Descrição
Apply(ADocument: TsgcPDFDocument)	Redige um documento carregado: ele é salvo, redigido e recarregado a partir dos bytes redigidos.
Apply(AInputFile, AOutputFile)	Redige um arquivo.
ApplyToStream(AInput, AOutput), ApplyToBytes(AData)	Redige um stream ou um array de bytes.

A saída é sempre um arquivo novo e completo (um salvamento completo): os bytes originais, que uma atualização incremental manteria, nunca são gravados. Depois de Apply, as estatísticas TextMatches, RemovedGlyphs, RedactedImages, RemovedImages, RemovedPaths e RemovedAnnotations ficam disponíveis, e `GetLog` retorna um log da operação.

Opções

Campo de TsgcPDFRedactionOptions	Descrição
FillColorR, FillColorG, FillColorB	Cor (0 a 1) dos retângulos pintados sobre as áreas. Padrão preto.
OverlayText, OverlayFontSize	Texto desenhado centralizado em cada área (Helvetica); 0 ajusta o texto à área.
RemoveAnnotationsInArea	Remove anotações cujo retângulo intersecta uma área, incluindo links e widgets de campos de formulário com seus campos. Padrão True.
RemoveMetadata	Remove o dicionário /Info e os metadados XMP.
RemoveAttachments	Remove arquivos incorporados e anotações de anexo de arquivo.
RemoveJavaScript	Remove JavaScript, /OpenAction, ações adicionais e as ações JavaScript, Launch, SubmitForm e ImportData.

Documentos Criptografados

Abra o documento com sua senha e chame `Apply(ADocument)`. O conteúdo é redigido descriptografado, e o documento mantém suas configurações de criptografia, de modo que o próximo SaveToFile ou SaveToStream o grava criptografado novamente. SaveIncremental gera uma exceção após uma redação. As sobrecargas de arquivo, stream e bytes rejeitam entrada criptografada, pois não recebem senha.

```
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;
```

Como o Conteúdo É Removido

- **Texto**: todo glifo cuja caixa intersecta uma área é removido do content stream; os glifos restantes mantêm suas posições.
- **Imagens**: imagens que o sgcPDF consegue decodificar, com 1 a 8 bits por componente, são reescritas com os pixels redigidos pintados na cor de preenchimento (RedactedImages). Imagens que não podem ser decodificadas, por exemplo com um filtro não suportado, grandes demais ou com 16 bits por componente, são removidas completamente (RemovedImages).
- **Gráficos vetoriais**: um subcaminho que está completamente dentro de uma área é removido (RemovedPaths). Subcaminhos que apenas cruzam uma área permanecem, sob o retângulo de preenchimento. Caminhos de recorte não são alterados.
- **Form XObjects** são redigidos recursivamente. Um form XObject aninhado profundamente demais ou com um filtro não suportado é removido por completo.
- Uma página cujo content stream usa um filtro não suportado gera EsgcPDFRedactionError em vez de ser deixada sem redação.

No exemplo, o pypdf não encontra nenhum vestígio das palavras redigidas na saída, nem no texto extraído nem em nenhum content stream, e o arquivo tem uma única revisão.

Veja também

- Criptografia por Certificado
- Extração de Texto

PDF Marcado

Um PDF marcado (ISO 32000-1, 14.8) contém uma árvore de estrutura lógica junto ao conteúdo da página: cabeçalhos, parágrafos, listas, tabelas, figuras com texto alternativo e links, na ordem de leitura. Leitores de tela, o reflow em dispositivos móveis e a exportação de texto o utilizam, e ele é exigido pela conformidade com PDF/UA e com o nível A do PDF/A.

Construindo a Estrutura

Defina **TsgcPDFDocument.Tagged** como True e envolva as chamadas de desenho de cada página entre `Page.Tagging.BeginElement` e `EndElement`. A marcação de página grava os operadores de conteúdo marcado (BDC/EMC com MCIDs) e os vincula ao elemento; `StructureTree` coleta os elementos, o ParentTree e o RoleMap, e grava /MarkInfo, /Lang e o /StructParents da página quando o documento é salvo.

```
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;
```

Membro	Descrição
Page.Tagging.BeginElement(AType; ATitle)	Abre um elemento cujo conteúdo é desenhado nesta página, como filho do elemento atual. Retorna o TsgcPDFStructElement.
StructureTree.BeginElement(AType; ATitle)	Abre um elemento de agrupamento sem conteúdo de página (Table, TR, L, Sect, Part...).
EndElement	Fecha o elemento atual (na marcação de página ou na árvore, correspondendo à chamada Begin).
BeginCustomElement(ATypeName...) and StructureTree.AddRole	Tipos de estrutura personalizados, mapeados para um tipo padrão pelo /RoleMap.
Page.Tagging.BeginArtifact(AKind; ASubtype), EndArtifact	Conteúdo que não faz parte do documento: cabeçalhos e rodapés repetidos, números de página, marcas d'água, decoração. AKind: akPagination, akLayout, akPage, akBackground. ASubtype: asHeader, asFooter, asWatermark.
StructureTree.Lang	Idioma natural do documento (catálogo /Lang), por exemplo en-US.

TsgcPDFStructType tem todos os tipos padrão: 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 e stAnnot.

Propriedades do Elemento

Propriedade	Descrição
Alt	Texto alternativo. Obrigatório para Figure e Formula: salvar um Figure sem Alt gera EsgcPDFTaggedError.
ActualText, Expansion, Lang, Title, ID	Texto de substituição, expansão de abreviação, idioma do elemento, título e identificador.
SetBBox(ALeft, ABottom, ARight, ATop)	Caixa delimitadora de um Figure, Formula ou Table.

Propriedade	Descrição
Scope, RowSpan, ColSpan, Headers	Células de tabela: escopo de um TH (tsRow, tsColumn, tsBoth), spans e os IDs das células de cabeçalho.

Links e Anotações

Uma anotação adicionada com `Annotations.AddLinkURI` (ou outro método Add) enquanto um elemento Link está aberto é anexada a ele como uma referência de objeto, com seu /StructParent. `Page.Tagging.AddAnnotation` anexa uma explicitamente.

Tabelas

Um TsgcPDFTable desenhado em um documento marcado é marcado automaticamente como Table, THead, TBody, TR, TH e TD. Com DrawMultiPage, as linhas de cabeçalho repetidas nas páginas seguintes são marcadas como artefatos de paginação, de modo que um leitor de tela as lê apenas uma 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;
```

Documentos marcados carregados mantêm sua árvore de estrutura ao serem salvos.

Veja também

- PDF/UA e PDF/A Nível A
- Validador de Acessibilidade
- Tabelas

PDF/UA e PDF/A Nível A

PDF/UA-1

PDF/UA-1 (ISO 14289-1) é o padrão para PDF universalmente acessível, exigido pelas leis de acessibilidade para documentos públicos em muitos países. Defina **TsgcPDFDocument.PDFUA** como True: o documento é salvo com pdfuaid:part 1 e dc:title nos metadados XMP e /DisplayDocTitle nas preferências do visualizador.
O salvamento gera uma exceção quando um requisito de PDF/UA não pode ser atendido:

- o documento não está marcado (`Tagged` é False);
- `StructureTree.Lang` está vazio;
- `Info.Title` está vazio;
- o texto é desenhado com uma das 14 fontes padrão, que não são incorporadas: use AddTrueTypeFont (ou AddTrueTypeFontFromFile) e DrawTextTT;
- o documento está criptografado sem a permissão ppExtract, que a tecnologia assistiva precisa.

PDF/A Nível A

O Nível A (acessível) do PDF/A adiciona os requisitos de PDF marcado ao nível B. Passe **paA1a**, **paA2a** ou **paA3a** para SetPDFALevel, e marque o conteúdo como no PDF/UA. PDF/UA e PDF/A podem ser combinados no mesmo arquivo:

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

O arquivo criado em PDF Marcado usa exatamente essa combinação e passa no veraPDF tanto para PDF/UA-1 quanto para PDF/A-2a.

TsgcPDFALevel	Padrão
paA1b, paA1a	PDF/A-1 (ISO 19005-1), nível B ou A
paA2b, paA2a	PDF/A-2 (ISO 19005-2), nível B ou A
paA3b, paA3a	PDF/A-3 (ISO 19005-3), nível B ou A. Permite qualquer arquivo incorporado.

Lista de Verificação

- Defina Info.Title e StructureTree.Lang.
- Incorpore todas as fontes (TrueType ou OpenType).
- Marque todo o conteúdo: cada texto e gráfico pertence a um elemento ou a um artefato.
- Dê a cada Figure e Formula um texto Alt.
- Use cabeçalhos em ordem (H1, depois H2...), células TH com Scope nas tabelas, e elementos Link para os links.
- Valide o arquivo salvo com o validador.

Veja também

- PDF Marcado
- Validador de Acessibilidade
- PDF/A

Validador de Acessibilidade

TsgcPDFAValidator (unidade sgcPDF_PDFA) verifica um arquivo salvo em relação a um nível de PDF/A. Para os níveis A, também verifica a estrutura, e com **PDFUA** definido como True, executa também as verificações 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;
```

O Que É Verificado

Método	Verificações
CheckStructure	Árvore de estrutura, /MarkInfo /Marked, /Lang.
CheckContentTagged	Todo operador de texto e de pintura de cada página pertence a um elemento de estrutura ou a um artefato.
CheckStructureElements	Texto alternativo de Figure e Formula, sequência de cabeçalhos, estrutura de tabelas e células de cabeçalho.
CheckFontUnicode	Toda fonte tem um mapa ToUnicode ou uma codificação padrão, para que o texto possa ser lido.
CheckPDFUA	pdfuaid:part, dc:title, /DisplayDocTitle, anotações na estrutura, /Tabs /S da página.

Validate executa todas as verificações do nível (e de PDF/UA quando ativado) e retorna True quando não há erros. Os métodos individuais podem ser chamados isoladamente.

Erros e Avisos

`Errors` lista os problemas que quebram a conformidade. Cada mensagem começa com a cláusula da norma à qual se refere, para que você possa consultá-la, por exemplo:

```
[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` lista problemas que não são erros da norma, mas prejudicam a acessibilidade, por exemplo:

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

O validador é uma verificação interna rápida para a sua própria saída. Ele não substitui um verificador de conformidade completo como o veraPDF ou o PAC para documentos de outras origens.

Veja também

- PDF/UA e PDF/A Nível A
- PDF Marcado
- PDF/A

Perguntas frequentes

Preciso implantar uma DLL?

Não. O sgcPDF é compilado no seu executável. Ele usa apenas bibliotecas que fazem parte do Windows: o GDI para renderização e carregamento de fontes, e a CryptoAPI para assinaturas digitais.

Quais plataformas são compatíveis?

Windows de 32 e 64 bits com Delphi 7 a Delphi 13 e C++Builder 2007 a 13. O núcleo, o renderizador raster e o visualizador FireMonkey também funcionam em Linux 64 bits, macOS, iOS e Android. Consulte Instalação.

Onde fica a origem das coordenadas da página?

No canto inferior esquerdo da página, em pontos (1/72 de polegada). O eixo Y cresce para cima. Consulte Arquitetura.

Por que caracteres acentuados ou não latinos não aparecem?

As fontes padrão cobrem apenas o conjunto de caracteres WinAnsi. Use uma fonte TrueType com AddTrueTypeFont e DrawTextTT. Consulte Texto e fontes.

Como escrevo texto em chinês, japonês ou coreano?

Adicione uma fonte TrueType CJK instalada, por exemplo o nome retornado por sgcPDFCJKFontNameFromEnum(cjkMSGothic), e desenhe com DrawTextTT.

Como abro um PDF protegido por senha?

Chame SetPassword antes de LoadFromFile. Consulte Criptografia.

Meu código de barras não aparece.

Passe o construtor de conteúdo da página (`Page.Content`) para TsgcPDFBarcodeGenerator, não a página. Consulte Códigos de barras.

sgcPDFExtractAllText retorna uma string vazia.

Um TsgcPDFParser precisa ser analisado depois de carregado: chame Parse depois de LoadFromFile. TsgcPDFDocument.LoadFromFile faz isso automaticamente.

Por que a assinatura é válida, mas não é confiável?

O certificado do assinante não encadeia com uma raiz confiável do Windows, o que é normal em certificados de teste autoassinados. Use um certificado emitido por uma autoridade certificadora confiável em produção.

O otimizador não reduziu meu arquivo.

Arquivos que já estão compactados e não têm objetos duplicados não podem ficar menores. Ative DownsampleImages ou as opções Remove para reduzi-los ainda mais. Consulte Otimizando arquivos PDF.

Posso limitar a memória usada ao abrir arquivos enviados?

Sim. Reduza sgcPDFMaxDecodedSize. Consulte Tratamento de erros e entradas hostis.

O sgcPDF funciona no Linux?

Sim. As unidades do núcleo compilam com o compilador Delphi para Linux de 64 bits: crie, edite, criptografe com senhas, extraia texto e renderize páginas em PNG com TsgcPDFRendererRaster em uma aplicação de servidor ou console. Carregue fontes de arquivos com AddTrueTypeFontFromFile. Consulte Linux.

Posso usar o sgcPDF a partir do C++Builder?

Sim. Pacotes para C++Builder 2007 a 13 estão incluídos e os cabeçalhos gerados já estão prontos para uso. Consulte C++Builder.

Como crio um arquivo PDF/A-3 com uma fatura incorporada (Factur-X, ZUGFeRD)?

Crie o XML da fatura no seu aplicativo e passe-o para Attachments.AddFacturX. O documento é salvo como PDF/A-3b com o XML e os metadados Factur-X. Consulte Faturas Factur-X e ZUGFeRD.

O sgcPDF pode criar assinaturas PAdES?

Sim. Defina SubFilter como pssCAdESDetached para PAdES B-B, depois adicione dados de validação com AddDSSDocument e um carimbo de tempo do documento com TimestampDocument para B-LT e B-LTA. Com o sgcSign, TsgcPDFPAdESSigner cria todos os níveis em uma única chamada com qualquer provedor de chaves do sgcSign. Consulte Assinaturas PAdES.

Um retângulo preto oculta o texto por baixo?

Não, o texto continua no arquivo. Use TsgcPDFRedactor, que remove o texto, os pixels de imagem e os gráficos das áreas redigidas. Consulte TsgcPDFRedactor.

Como faço um PDF acessível?

Marque o conteúdo com Page.Tagging, defina Info.Title e StructureTree.Lang, incorpore as fontes e defina PDFUA como True. Consulte PDF Marcado.

Introdução

O sgcReports é um engine de relatórios headless e neutro de plataforma para Delphi, construído sobre sgcPDF. Ele visa, em primeiro lugar, a geração de relatórios em servidor e na web: serviços Windows, daemons Linux64 e endpoints HTTP, e roda sem alterações em aplicações desktop e móveis VCL e FMX.
O engine produz, por padrão, saída em PDF acessível (PDF/UA-1) e PDF arquivístico (PDF/A-3). Um relatório também pode ser exportado para XLSX, CSV ou PNG a partir das mesmas páginas preparadas.

Headless por design

O núcleo (modelo de relatório, expressões, fontes de dados, fontes, layout, engine e exports) não tem dependência de VCL ou FMX nem de DLL externa. Ele compila e roda em Windows, Linux64, macOS, iOS e Android. Os componentes de preview (`TsgcReportPreviewVCL`, `TsgcReportPreviewFMX`) e o viewer de navegador são pacotes complementares opcionais.

Acessível e arquivístico por padrão

A exportação em PDF marca a árvore de estrutura automaticamente: títulos, parágrafos, tabelas com linhas de cabeçalho, figuras com texto alternativo e artefatos, de modo que a propriedade `PDFUA` é `True` por padrão. A saída arquivística PDF/A-3 é controlada pela propriedade `PDFA` e pode ser combinada com PDF/UA. Veja Acessibilidade e PDF/A.

Plataformas

Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Windows, Linux64, macOS, iOS e Android através do FireMonkey. O núcleo compila no Delphi 7 sem generics, métodos anônimos, variáveis inline ou métodos de record.

Edições e requisitos

Pacote	Requer	Necessário para
`sgcReports` (núcleo)	sgcPDF	Modelo de relatório, expressões, fontes de dados, fontes, layout, engine, export para PDF/XLSX/CSV/PNG, API fluente, definições em JSON. O sgcPDF é obrigatório em toda configuração.
`sgcReportsVCL` / `sgcReportsFMX`	núcleo, VCL ou FMX	`TsgcReportPreviewVCL` ou `TsgcReportPreviewFMX`.
`sgcReportsHTML`	núcleo, sgcWebSockets (edição Professional ou superior), sgcHTML	`TsgcReportHTTPEndpoint`, `TsgcReportHTMLViewer` e `TsgcReportHTMLDesigner`. O sgcWebSockets é necessário somente para este pacote; nunca é usado pelo núcleo.

Próximo: Instalação, depois Primeiros passos.

Instalação

O sgcReports faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença. O sgcReports requer o sgcPDF, e o instalador cuida disso.

Requisitos

- Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 ou 13 (ou a versão correspondente do C++Builder, C2007 a C13).
- sgcPDF, mesma versão do Delphi. O instalador do sgcUI o instala junto com o sgcReports.
- sgcWebSockets edição Professional ou superior, e sgcHTML, apenas quando o pacote `sgcReportsHTML` é instalado.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcReports (installs sgcPDF)** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcReports

- **sgcPDF**, o engine sobre o qual o sgcReports é construído. Selecionar o sgcReports também seleciona o sgcPDF, e o sgcPDF é instalado sempre que o sgcReports é.
- **sgcReports HTML for sgcWebSockets**, listado em **sgcWebSockets bridges**. Instala os pacotes `sgcReportsHTML` (endpoint HTTP, viewer de navegador e designer web) para o Delphi 10 Seattle e posteriores. O instalador só o oferece quando o sgcWebSockets já está instalado na máquina e a sua licença inclui o sgcReports. Caso contrário, ele aparece esmaecido.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado, incluindo o sgcReports e o sgcPDF. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13`.
`Demos\Reports`	Os demos do sgcReports.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcPDF fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcReports e o sgcPDF. Uma compilação trial mostra um aviso na inicialização (`sgcReports_License.pas`). Uma compilação licenciada não mostra.
- **Community**: o sgcPDF não tem edição Community, portanto o instalador gratuito sgcUI Community não o instala. Ele oferece o sgcReports apenas quando um sgcPDF licenciado já está instalado na máquina, seja de uma instalação registrada do sgcUI que inclui o sgcPDF, seja de uma instalação autônoma do sgcPDF, e compila o sgcReports com esse sgcPDF. Sem um deles, o sgcReports aparece esmaecido e o instalador pede que você instale o sgcPDF primeiro. As pontes sgcWebSockets não fazem parte do instalador Community.

Migrando de instalações autônomas do sgcPDF e do sgcReports

Se o sgcPDF ou o sgcReports foi instalado antes com o seu próprio instalador autônomo, o instalador do sgcUI detecta essa instalação e lista os library paths e os pacotes de design dela na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho. A página só é exibida quando uma instalação antiga é encontrada. Após a migração, os dois produtos são compilados a partir da única pasta `sgcUI\Source`.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcReportsD13.dpk`.

Pacote	Conteúdo
`sgcPDF<V>` , `sgcPDFVCL<V>` , `sgcPDFFMX<V>` / `dclsgcPDF<V>` , `dclsgcPDFFMX<V>`	sgcPDF, o engine que o sgcReports requer (runtime / design time).
`sgcReports<V>`	Engine e exports do núcleo (runtime).
`sgcReportsVCL<V>` / `dclsgcReports<V>`	Preview VCL (runtime / design time).
`sgcReportsFMX<V>` / `dclsgcReportsFMX<V>`	Preview FMX (runtime / design time).
`sgcReportsHTML<V>` / `dclsgcReportsHTML<V>`	Endpoint HTTP, viewer de navegador e designer web (runtime / design time).

Compile-os na ordem de dependência: primeiro os pacotes de runtime do sgcPDF, depois o núcleo do sgcReports, depois VCL ou FMX, depois os pacotes de design time, e `sgcReportsHTML` por último. Em seguida, instale os pacotes de design time `dcl*`. `TsgcReport` aparece na página da paleta **sgcReports** depois de instalar o pacote de design time.

Search path

Adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path), ou adicione-o ao unit search path do projeto. O sgcReports e o sgcPDF compartilham essa pasta, portanto uma única entrada cobre os dois.

C++Builder

Abra o projeto `.cbproj` correspondente à sua versão do C++Builder em `sgcUI\Packages`, por exemplo `sgcReportsC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

Primeiros Passos

Existem três maneiras de construir um relatório: a API de modelo (preencher um componente TsgcReport com páginas, bandas e itens), a API fluente (encadear chamadas em um TsgcReportBuilder), ou uma definição de relatório em JSON carregada em tempo de execução. As três terminam no mesmo TsgcReportEngine e nos mesmos exports.

1. A API de modelo

Coloque um `TsgcReport`, um `TsgcReportEngine` e um `TsgcReportPDFExport` em um formulário (ou crie-os em código) e preencha o relatório com uma página, uma banda e um item 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. A API fluente

O mesmo relatório com `TsgcReportBuilder`, que não precisa de nenhum componente no formulário:

```
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;
```

Toda chamada de estilo, página e banda retorna um builder com escopo próprio (`TsgcReportStyleBuilder`, `TsgcReportPageBuilder`, `TsgcReportBandBuilder`); `EndStyle`, `EndBand` e `EndPage` retornam ao escopo envolvente para que a cadeia possa continuar.
Veja TsgcReportBuilder para a cadeia completa (estilos, páginas, bandas, itens, dados, helpers de export).

3. Uma definição de relatório em JSON

Um relatório construído de qualquer uma das formas pode ser salvo como JSON e recarregado sem 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;
```

Veja Definições de relatório em JSON para o formato do envelope, as regras de nomenclatura e o schema gerado.

Próximos passos

- Modelo de relatório: páginas, bandas, itens, estilos e units.
- Fontes de dados: ler registros de um `TDataSet`, de JSON ou de eventos.
- Como fazer: fatura com dados mestre-detalhe.

Modelo de Relatório

Um relatório (TsgcReport) é uma árvore de páginas, bandas, itens e estilos, construída sobre `TCollection`, de modo que é transmitido em stream como um DFM, como qualquer outro componente. A mesma árvore também pode ser carregada e salva em JSON, veja Definições de relatório em JSON.

Units

`TsgcReport.Units` (`TsgcReportUnits`: `ruMillimeters`, `ruPoints`, `ruInches`) é a unidade em que cada `Left`, `Top`, `Width`, `Height` e margem é informado. Internamente, o layout e a exportação em PDF sempre trabalham em pontos (1/72 de polegada); o relatório converte na leitura e na gravação dessas propriedades.

Pages

`TsgcReportPage` (em `TsgcReport.Pages`) tem `PaperSize` (`rpsA3`, `rpsA4`, `rpsA5`, `rpsLetter`, `rpsLegal`, `rpsCustom`), `Orientation`, `Width` e `Height` (usados somente quando `PaperSize` é `rpsCustom`), `Margins`, `Columns` e `ColumnSpacing`. Um relatório pode ter mais de uma definição de página; os subrelatórios `rbtDetailData` referenciam uma página pelo nome (`TsgcReportItem.SubreportPageName`).

Bandas

`TsgcReportBand` (em `TsgcReportPage.Bands`) tem um `BandType` que decide quando e com que frequência ela é impressa:

BandType	Impressão
rbtReportTitle	Uma vez, antes do primeiro cabeçalho de página.
rbtReportSummary	Uma vez, após o último grupo e os dados de detalhe.
rbtPageHeader	No topo de toda página.
rbtPageFooter	No rodapé de toda página.
rbtColumnHeader	No topo de toda coluna, em uma página multicoluna.
rbtColumnFooter	No rodapé de toda coluna.
rbtGroupHeader	Uma vez para cada valor distinto de `GroupExpression`, com `RepeatOnEveryPage` opcionalmente repetindo-a quando um grupo atravessa uma quebra de página.
rbtGroupFooter	Uma vez ao final de cada grupo, após seus agregados serem concluídos.
rbtMasterData	Itera os registros de `DataSourceName`; não imprime nada por si mesma.
rbtDetailData	Itera os registros de `DataSourceName` sob o registro mestre atual.
rbtChild	Impressa imediatamente após sua banda pai através de `ChildBandName`, compartilhando o mesmo registro.

Outras propriedades de banda: `Height`, `CanGrow` e `CanShrink` (crescer ou encolher para se ajustar ao seu item mais alto), `KeepTogether` (não dividir a banda por uma quebra de página), `StartNewPage`, `PrintOnFirstPage` / `PrintOnLastPage`, `VisibleExpression` e `StyleName` (um estilo padrão para seus itens).

Itens e tipos

`TsgcReportItem` (em `TsgcReportBand.Items`) tem um `Kind` (`TsgcReportItemKind`):

Kind	Subopções	Notas
rikText	`Text` (`TsgcReportTextOptions`)	`Text.Text` é um modelo de texto, veja Expressões e modelos de texto.
rikImage	`Image` (`TsgcReportImageOptions`)	Bytes de imagem incorporada, transmitidos em stream como `PictureData`.
rikLine	—	Uma régua horizontal ou vertical.
rikShape	`Shape` (`TsgcReportShapeOptions`)	`ShapeKind`: retângulo, retângulo arredondado, elipse, linha.
rikBarcode	`Barcode` (`TsgcReportBarcodeOptions`)	11 simbologias, veja `TsgcReportBarcodeSymbology`; `Barcode.Value` é um modelo de texto.
rikSubreport	SubreportPageName	Executa, no próprio local, outra página do mesmo relatório.
rikPageBreak	—	Força uma nova página.

Propriedades comuns de item: `Left`, `Top`, `Width`, `Height`, `StyleName`, `Visible` / `VisibleExpression`, `CanGrow` / `CanShrink`, `AltText` (acessibilidade) e `TagRole` (veja Acessibilidade).

Estilos

`TsgcReportStyle` (em `TsgcReport.Styles`) agrupa `Font`, `BackColor`, `Frame`, `Padding`, `HAlign` e `VAlign` sob um `Name` que uma banda ou um item referencia por `StyleName`. As cores são `TsgcReportColor`, um cardinal `$00RRGGBB` neutro de plataforma (não `Graphics.TColor`); `sgcReportColorNone`, `sgcReportColorBlack` e `sgcReportColorWhite` são predefinidas.

Expressões e Modelos de Texto

O sgcReports compila cada expressão uma vez, depois a avalia uma vez por registro. Os valores são Variants.

Campos, variáveis e escopo

Um campo é escrito `Source.Field`, onde `Source` é o `Name` de uma fonte de dados adicionada ao engine (`TsgcReportEngine.AddDataSource`) e `Field` é um de seus nomes de campo. Um identificador isolado é resolvido como uma variável do engine (`TsgcReportEngine.Variables`, pares `Name=Value`), também acessível a partir de `OnGetVariable`. Dois identificadores estão sempre disponíveis: `Page` (número da página atual) e `TotalPages`, resolvido somente após a última página (adiado, em duas passagens, "Página X de Y").
Um rodapé de grupo ou um rodapé de página vê os valores de campo do **último** registro de seu grupo ou página, não o próximo registro que disparou a quebra.

Operadores

Aritméticos `+ - * /`, de comparação `= <> < <= > >=`, lógicos `and or not`, concatenação de strings com `+`, e parênteses para agrupamento.

Funções

Função	Descrição
IIF(cond, a, b)	a quando cond é verdadeiro, senão b.
IsNull(v)	True quando v é Null ou vazio.
Coalesce(v1, v2, ...)	Primeiro argumento que não é Null.
Upper(s)	Maiúsculas.
Lower(s)	Minúsculas.
Trim(s)	Remove espaços à esquerda e à direita.
Length(s)	Comprimento da string.
Copy(s, from, count)	Substring.
Pos(sub, s)	Posição baseada em 1, 0 quando não encontrado.
Replace(s, from, to)	Substitui todas as ocorrências.
FormatFloat(fmt, v)	Formata um número.
FormatDateTime(fmt, v)	Formata uma data ou hora.
Str(v)	Converte para string.
Round(v)	Arredonda para o número inteiro mais próximo.
Trunc(v)	Trunca em direção a zero.
Abs(v)	Valor absoluto.

Int(v)	Parte inteira.
MinOf(a, b)	O menor dos dois.
MaxOf(a, b)	O maior dos dois.
Now()	Data e hora atuais.
Date()	Data atual.
Year(v)	Ano de uma data.
Month(v)	Mês de uma data.
Day(v)	Dia de uma data.

Agregados

`Sum`, `Count`, `Avg`, `Min` e `Max` recebem um argumento, por exemplo `Sum(Lines.Qty * Lines.Price)`. Um agregado acumula ao longo do escopo em que é impresso: dentro de um rodapé de grupo ele reinicia a cada grupo, dentro de um rodapé de página ele reinicia a cada página, dentro do resumo do relatório ele é executado para o relatório inteiro.

Semântica de Null

Um campo ausente ou uma variável não definida é avaliado como Null, que se propaga através da aritmética (`Null + 1` é Null) e é comparado como nem igual nem diferente de qualquer valor não Null, seguindo as regras de Variant do Delphi. Use `IsNull` ou `Coalesce` para substituir por um valor padrão antes da formatação.

Modelos de texto

`TsgcReportTextTemplate` (usado por `TsgcReportItem.Text.Text` e `TsgcReportItem.Barcode.Value`) é texto literal com expressões incorporadas entre colchetes, por exemplo `'Invoice [Invoices.Number], total [Sum(Lines.Qty * Lines.Price)]'`. Tudo fora de `[ ]` é copiado como está, de modo que um valor de código de barras como `STOCK-2026-09`, sem colchetes, é impresso literalmente, zeros à esquerda incluídos. `TsgcReportTextOptions.DisplayFormat` (ou `Barcode.DisplayFormat`, onde aplicável) formata o resultado avaliado de um modelo de expressão única do mesmo modo que `FormatFloat` ou `FormatDateTime` fariam.

Fontes de Dados

Uma fonte de dados é uma descendente de `TsgcReportDataSource`, adicionada a um TsgcReportEngine com `AddDataSource`. Sua propriedade `SourceName` (não o `Name` do componente) é o que as expressões de banda referenciam como `Source.Field`, e uma banda `rbtMasterData` ou `rbtDetailData` a referencia pelo mesmo nome através de `DataSourceName`.

Tipos

Classe	Lê a partir de
TsgcReportDataSetSource	Qualquer `TDataSet` (`TClientDataSet` do VCL, FireDAC, dbExpress, ...).
TsgcReportJSONSource	Uma matriz JSON de objetos, a partir de uma string, um arquivo ou um stream.
TsgcReportEventSource	Handlers de evento `OnFirst` , `OnNext` , `OnEof` e `OnGetValue` , para dados que não têm forma de dataset nem de JSON, como uma chamada REST paginada.

Mestre-detalhe

`TsgcReportDataSource` (a base comum) tem `MasterSource`, `MasterFields` e `DetailFields` (separados por ponto e vírgula, mesma quantidade nos dois lados). Quando uma banda de detalhe com `DataSourceName` apontando para a fonte de detalhe é aninhada sob uma banda mestre, o engine reinicia a fonte de detalhe e a associa ao registro mestre atual por esses campos antes de cada iteração de detalhe. É assim que a demo de fatura vincula as linhas da fatura à sua fatura, veja Como fazer: fatura com dados mestre-detalhe.

Registros JSON

`TsgcReportJSONSource.RecordsPath` é um caminho separado por pontos até a matriz de registros; deixe-o vazio quando a própria raiz do JSON for a matriz. Os nomes de campo são descobertos a partir dos membros do primeiro registro.

Segurança em threads

Uma fonte de dados não é compartilhada entre relatórios preparados simultaneamente; cada thread deve usar suas próprias instâncias de fonte (e seu próprio engine). Veja Segurança em threads e uso em servidor.

Grupos, Cabeçalhos, Rodapés e Colunas

Grupos

Uma banda `rbtGroupHeader` com uma `GroupExpression` (por exemplo `'Sales.Region'`) inicia um novo grupo sempre que o valor dessa expressão muda; sua banda `rbtGroupFooter` correspondente encerra o grupo depois que seus agregados são finalizados. Defina `RepeatOnEveryPage` no cabeçalho para reimprimi-lo quando o grupo atravessar uma quebra de página, e `StartNewPage` para forçar cada grupo em sua própria página (usado para uma página por fatura). Os grupos se aninham: uma página pode ter mais de um par `rbtGroupHeader` / `rbtGroupFooter`, avaliados na ordem em que as bandas aparecem.

Cabeçalhos e rodapés de página e de coluna

`rbtPageHeader` e `rbtPageFooter` são impressos em toda página. `rbtColumnHeader` e `rbtColumnFooter` são impressos em toda coluna quando `TsgcReportPage.Columns` é maior que 1. `PrintOnFirstPage` e `PrintOnLastPage` (padrão `True`) suprimem um cabeçalho ou rodapé de página na primeira ou na última página.

Manter unido (Keep-together)

`TsgcReportBand.KeepTogether := True` força a banda inteira para uma única página: se ela não couber no espaço restante da página atual, o engine inicia uma nova página antes de imprimi-la, em vez de dividi-la. Usado em blocos de totais e outras bandas que não podem ser cortadas por uma quebra de página.

Colunas

`TsgcReportPage.Columns` (padrão 1) e `ColumnSpacing` dispõem as bandas mestre e detalhe em colunas estilo jornal; o engine preenche uma coluna antes de passar para a próxima, imprimindo `rbtColumnHeader` / `rbtColumnFooter` em cada limite de coluna.

Subrelatórios

Um item `rikSubreport` (`TsgcReportItem.SubreportPageName`) executa, no próprio local, outra `TsgcReportPage` do mesmo relatório, na posição do item. A página do subrelatório pode ter suas próprias bandas e suas próprias referências a fontes de dados, e é preparada como parte da mesma execução.

Página X de Y

`Page` e `TotalPages` são os dois identificadores reservados para o texto de paginação, tipicamente `'Page [Page] of [TotalPages]'` em uma banda `rbtPageFooter`. `TotalPages` é adiado: o engine executa o relatório inteiro uma vez para contar as páginas, depois resolve toda expressão adiada em uma segunda passagem antes de as páginas serem entregues a um export.

Fontes

`TsgcReportFontManager` (unidade `sgcReports_Fonts`) descobre, mede e aplica fallback de fontes tanto para o engine de layout (quebra de linha, `CanGrow`) quanto para a exportação em PDF. Ele não tem estado global; cada engine e cada export referenciam um manager através de sua propriedade `FontManager`.

Manager padrão

Quando `FontManager` é deixado sem atribuição, o engine e os exports usam o `sgcReportDefaultFontManager`, uma instância compartilhada em todo o processo, construída a partir das fontes instaladas no sistema e criada uma vez, no primeiro uso. Atribua seu próprio `TsgcReportFontManager` quando um relatório precisar de fontes que o manager padrão não possui, ou para manter o carregamento de fontes isolado entre relatórios concorrentes.

Descoberta

O manager varre as pastas de fontes da plataforma: a pasta Fonts do Windows, os diretórios padrão de fontes do Linux, os locais de fontes do sistema em macOS e iOS, e os caminhos de fontes e assets do Android. As faces TrueType descobertas são indexadas por nome de família, peso e estilo (negrito, itálico).

Fontes padrão e fallback

As 14 fontes PDF padrão (Helvetica, Times, Courier e suas variantes em negrito e itálico, além de Symbol e ZapfDingbats) são mapeadas por padrão; um estilo com `Font.Name := 'Helvetica'` é renderizado sem incorporação, a menos que a exportação exija uma face incorporável (PDF/A, ou um caractere que a fonte base não consiga representar). O manager então substitui uma face TrueType para a família solicitada, e recorre ainda mais, glifo a glifo, a outra face instalada quando a face atual não tem um glifo para um caractere, o que é o que permite que uma fonte base latina ainda renderize texto CJK ou cirílico presente nos dados.

Incorporação

A exportação em PDF incorpora as faces TrueType que realmente usou, em subconjunto restrito aos glifos referenciados pelo documento, exceto quando o PDF/A exige a incorporação completa de uma face já parcialmente incorporada. `TsgcReportPDFExport.FontManager` resolve as faces dos itens sem face explícita e, para a saída PDF/A, os substitutos TrueType das 14 fontes padrão através de `TsgcReportFontManager.ResolveEmbeddable`.

Plataformas móveis e headless

O iOS não tem uma pasta de fontes do sistema legível a partir de um aplicativo em sandbox, e um daemon Linux64 pode ser executado em uma imagem mínima com poucas fontes instaladas. Em ambos os casos, distribua uma fonte TrueType (por exemplo DejaVu Sans, cuja licença é livre de copyleft) junto ao executável ou dentro do pacote do aplicativo e deixe a descoberta de fontes recorrer a ela; veja Como fazer: um relatório FMX móvel.

Segurança em Threads e Uso em Servidor

O sgcReports é projetado para geração de relatórios em servidor e na web, onde muitos relatórios são preparados e exportados ao mesmo tempo em threads diferentes.

Regra

Um engine, seu relatório, suas fontes de dados e seus exports não são compartilhados entre threads. Cada thread de trabalho deve criar e usar seu próprio `TsgcReportEngine`, `TsgcReport` (ou compartilhar uma definição de relatório somente leitura e clonar apenas o engine, já que o engine mantém o estado da execução), suas próprias instâncias de fonte de dados e seus próprios componentes de export. Nada nas classes de modelo, layout ou export usa um lock global em torno do estado próprio de um relatório, portanto duas threads compartilhando um engine ou um export corrompem ambas as execuções.

O que pode ser compartilhado

Um `TsgcReportFontManager` pode ser compartilhado somente leitura entre threads assim que suas fontes são descobertas, incluindo o `sgcReportDefaultFontManager` de escopo do processo: a descoberta e a medição de fontes não alteram um estado por chamada do qual outras threads dependam. `TsgcReportFontManager.ResolveEmbeddable` também não altera a configuração `EmbedStandardFonts` do gerenciador, portanto também é seguro chamá-lo em um gerenciador compartilhado entre threads. Uma `TsgcReportDefinition` (ou as funções simples `sgcReportLoadFromJSON` / `sgcReportSaveToJSON`) pode ser chamada de qualquer thread, desde que cada chamada opere em sua própria instância de `TsgcReport`.

Padrão: um pool de workers

Carregue uma única definição de relatório em JSON compartilhada e somente leitura na inicialização. Cada thread de trabalho então faz, para cada job: cria um `TsgcReport`, chama `sgcReportLoadFromJSON` nele, cria um `TsgcReportEngine` e as fontes de dados daquele job, adiciona-as, `Prepare`, exporta, e então libera tudo o que a thread criou. Este é o design usado pela demo do daemon Linux, verificado com um teste de estresse de 8 threads em Win32, Win64 e Linux64.

Aplicações de servidor no Windows

`sgcReportsConfigureForServer` (unidade `sgcReports_Types`) é uma configuração opcional para servidores que preparam e exportam relatórios em muitas threads ao mesmo tempo. No Windows, com Delphi 2006 ou posterior, ela define `System.NeverSleepOnMMThreadContention` como `True`, de modo que o gerenciador de memória do RTL gira em espera ativa em vez de dormir quando duas threads disputam o mesmo tamanho de bloco; ela não tem efeito em outras plataformas nem em compiladores mais antigos. O sgcReports nunca a chama, portanto ela não tem efeito até que a própria aplicação a chame, e quando isso acontece, a configuração se aplica a todo o processo, a cada alocação de memória feita pela aplicação, não apenas às exportações de relatórios.
Chame-a uma única vez, na inicialização da aplicação, antes que qualquer thread de trabalho seja criada, em um servidor multithread. Medido com 8 threads exportando 25 relatórios cada no Win64, o throughput subiu de cerca de x1,8 para cerca de x5 em relação a uma única thread, em duas execuções (de 13,6 s para 4,4 s, e de 7,8 s para 2,8 s). A configuração também faz as threads usarem mais CPU enquanto esperam, portanto deixe-a desativada em aplicações de thread única.

```
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 um engine de vida longa por nome de relatório e atende requisições concorrentes para ele; veja o tópico desse componente para como ele sincroniza o acesso ao engine compartilhado e ao seu cache de páginas preparadas.

Acessibilidade (PDF/UA)

TsgcReportPDFExport.PDFUA é `True` por padrão: todo PDF exportado é marcado (tagged) para PDF/UA-1 (ISO 14289-1), a menos que isso seja desativado explicitamente. O tagging grava uma árvore de estrutura (títulos, parágrafos, tabelas com linhas de cabeçalho, figuras com texto alternativo, artefatos), o `/Lang` do documento, o título e o identificador `pdfuaid`, e incorpora todas as fontes usadas (as fontes padrão são substituídas por uma fonte TrueType equivalente através do font manager, já que um PDF marcado não pode depender de uma fonte base não incorporada).

Regras de tagging automático

O papel de cada item na árvore de estrutura é derivado de sua banda e tipo, a menos que seja sobrescrito:

Origem	Papel padrão
Item de texto em uma banda `rbtReportTitle`	Título (o nível depende de `TagRole`, se definido; caso contrário, um título de nível superior).
Item de texto em uma banda `rbtGroupHeader`	Título.
Outros itens de texto	Parágrafo.
Uma tabela construída a partir de linhas de detalhe repetidas sob uma banda de cabeçalho de coluna	Tabela, com as células da banda de cabeçalho de coluna marcadas como células de cabeçalho de tabela.
Itens de imagem e código de barras	Figura, usando `AltText` como o texto alternativo.
Linhas e formas decorativas	Artefato (excluído da ordem de leitura apresentada à tecnologia assistiva).

TagRole

Defina `TsgcReportItem.TagRole` para sobrescrever o papel automático: `rtrAuto` (padrão), `rtrHeading1`, `rtrHeading2`, `rtrHeading3`, `rtrParagraph`, `rtrFigure`, `rtrArtifact`, `rtrTableHeader` ou `rtrTableCell`. Defina sempre `AltText` em um item marcado como `rtrFigure`, já que o texto alternativo é o que o PDF/UA exige para que uma figura seja aprovada.

Ordem de leitura

A árvore de estrutura segue a ordem em que os itens são impressos, de cima para baixo, da esquerda para a direita dentro de uma banda, o que, para um layout de relatório normal, já corresponde à ordem de leitura pretendida; não há uma etapa manual separada de ordenação.

Verificando com o veraPDF

Veja Como fazer: um PDF acessível validado com o veraPDF para o comando que verifica um arquivo PDF/UA-1 em relação ao perfil ISO.

PDF/A

TsgcReportPDFExport.PDFA (`TsgcReportPDFAMode`) seleciona a saída arquivística: `rpaNone` (PDF simples, padrão), `rpaA3b` (PDF/A-3b, reprodutibilidade visual) ou `rpaA3a` (PDF/A-3a, adiciona tagging e requisitos de idioma além do A-3b). O PDF/A-3 permite incorporar arquivos arbitrários no documento, o que o torna a escolha arquivística para nota fiscal eletrônica e fluxos semelhantes, em que a fonte legível por máquina (por exemplo o XML da fatura) viaja dentro do mesmo PDF.

Combinando com PDF/UA

`PDFA = rpaA3a` e `PDFUA = True` podem ser definidos juntos: ambos exigem uma árvore de estrutura marcada, um idioma declarado e fontes incorporadas, de modo que a exportação produz uma única árvore de estrutura que satisfaz as duas conformidades. Esta é a combinação usada para faturas e outros documentos que devem ser ao mesmo tempo acessíveis e arquiváveis; veja o como fazer da fatura e o como fazer do veraPDF.

Fontes

O PDF/A exige que toda fonte seja incorporada. Quando um estilo usa uma das 14 fontes padrão pelo nome, a exportação substitui e incorpora uma face TrueType através do font manager, em vez de depender da fonte base, o que o PDF/A proíbe como única representação do texto; veja Fontes.

Saída reproduzível

`FixedCreationDate` e `FixedDocumentID` em `TsgcReportPDFExport` fixam as datas de criação e modificação e o identificador do arquivo a um valor fornecido pelo chamador em vez da hora atual, de modo que exportações repetidas do mesmo relatório produzem arquivos byte a byte idênticos; útil para testes de golden-file e deduplicação de arquivamento. Definir qualquer um dos dois desativa os object streams do PDF/A para que os valores permaneçam corrigíveis.

TsgcReport

Unidade: sgcReports_Model
A definição do relatório: páginas, bandas, itens e estilos. Baseado em `TComponent` e construído sobre `TCollection`, portanto é transmitido em stream como um DFM, como qualquer outro componente, e também pode ser carregado ou salvo em JSON, veja Definições de relatório em JSON. Veja Modelo de relatório para a descrição completa de páginas, bandas, itens e estilos.

Propriedades

Propriedade	Tipo	Descrição
Units	TsgcReportUnits	Unidade em que cada propriedade de geometria (Left, Top, Width, Height, margens) é informada: ruMillimeters, ruPoints ou ruInches.
Pages	TsgcReportPages	Collection de TsgcReportPage.
Estilos	TsgcReportStyles	Collection de TsgcReportStyle, referenciada por StyleName a partir de bandas e itens.
Title	string	Título do relatório; também o título padrão do documento PDF.
Author	string	Autor do relatório.

Exemplo

Consulte Primeiros passos para um exemplo completo da API de modelo (estilos, uma página, uma banda, um item de texto) e Como fazer: fatura com dados mestre-detalhe para um relatório completo.

TsgcReportEngine

Unidade: sgcReports_Engine
Executa um TsgcReport contra suas fontes de dados: percorre relações mestre-detalhe e grupos, avalia expressões e agregados, e produz a lista de exibição de páginas preparadas que os exports e os componentes de preview leem. Baseado em `TComponent`.

Propriedades

Propriedade	Tipo	Descrição
Report	TsgcReport	A definição do relatório a ser executada.
Variables	TStrings	Parâmetros do usuário como Name=Value, expostos às expressões como identificadores simples.
FontManager	TsgcReportFontManager	Não published (public). Nil usa o sgcReportDefaultFontManager compartilhado. Veja Fontes.
OnBeforePrintBand	TsgcReportBeforePrintBandEvent	Disparado antes de uma banda ser impressa; permite ao handler alterar visibilidade ou valores.
OnGetVariable	TsgcReportGetVariableEvent	Resolve um identificador não encontrado em Variables.

Métodos

Método	Retorno	Descrição
AddDataSource(aSource: TsgcReportDataSource)	—	Registra uma fonte de dados sob seu SourceName; as expressões de banda passam então a referenciá-la como SourceName.Field.
RemoveDataSource(aSource: TsgcReportDataSource)	—	Remove o registro de uma fonte de dados.
ClearDataSources	—	Remove o registro de todas as fontes de dados.
Prepare	TsgcReportPreparedPages	Executa o relatório; o chamador assume a posse e libera o resultado.
Prepare(aPages: TsgcReportPreparedPages)	—	Sobrecarga que prepara dentro de um TsgcReportPreparedPages existente e previamente limpo.

Exemplo

```
sgcEngine.Report := sgcReport;
sgcEngine.AddDataSource(srcInvoices);
sgcEngine.AddDataSource(srcLines);
oPages := sgcEngine.Prepare;
try
  pdfExport.SaveToFile(oPages, 'invoice.pdf');
```

```
finally
  oPages.Free;
end;
```

TsgcReportDataSetSource

Unidade: sgcReports_Data
Fonte de dados que lê registros de qualquer `TDataSet` (`TClientDataSet`, FireDAC, dbExpress, entre outros). Descende de `TsgcReportDataSource`, que fornece as propriedades comuns de mestre-detalhe `MasterSource`, `MasterFields` e `DetailFields`, veja Fontes de dados.

Propriedades

Propriedade	Tipo	Descrição
DataSet	TDataSet	O dataset a ser lido. Se deixado fechado, é aberto e fechado pela fonte a cada execução; se já estiver aberto, seu estado e posição são restaurados depois.
SourceName	string	Herdado. O nome que as expressões de banda usam como Source.Field; independente do Name do componente.
MasterSource	TsgcReportDataSource	Herdado. A fonte mestre da qual esta é detalhe.
MasterFields	string	Herdado. Nomes dos campos mestres separados por ponto e vírgula.
DetailFields	string	Herdado. Nomes dos campos de detalhe separados por ponto e vírgula, na mesma quantidade que MasterFields.

Exemplo

```
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 Como fazer: fatura com dados mestre-detalhe para o relatório completo que usa este par de fontes.

TsgcReportJSONSource

Unidade: sgcReports_Data
Fonte de dados que lê registros de uma matriz JSON de objetos, informada como uma string, um nome de arquivo ou um stream. Descende de `TsgcReportDataSource`, portanto também suporta `MasterSource`, `MasterFields` e `DetailFields`.

Propriedades

Propriedade	Tipo	Descrição
JSON	string	O texto JSON, quando definido diretamente.
FileName	string	Um arquivo do qual ler o JSON.
RecordsPath	string	Caminho separado por pontos até a matriz de registros; vazio quando a própria raiz do JSON é a matriz.

Métodos

Método	Descrição
LoadFromStream(aStream: TStream)	Analisa o JSON UTF-8 do stream imediatamente; substitui JSON e FileName até que um dos dois mude.

Exemplo

sales.json, uma matriz sob "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 Como fazer: um relatório agrupado com totais.

TsgcReportEventSource

Unidade: sgcReports_Data
Fonte de dados controlada inteiramente por eventos, para registros que não têm forma de `TDataSet` nem de JSON, por exemplo uma resposta REST paginada obtida sob demanda. Sem um handler `OnEof` a fonte não tem registros.

Propriedades

Propriedade	Tipo	Descrição
FieldNames	TStrings	Os nomes de campo que esta fonte expõe, já que não há dataset nem objeto JSON de onde descobri-los.
OnFirst	TNotifyEvent	Disparado para reiniciar a iteração no primeiro registro.
OnNext	TNotifyEvent	Disparado para avançar ao próximo registro.
OnEof	TsgcReportEofEvent	`procedure(Sender: TObject; var aEof: Boolean) of object`. Defina aEof como True quando não houver mais registros.
OnGetValue	TsgcReportGetValueEvent	`procedure(Sender: TObject; const aFieldName: string; var aValue: Variant) of object`. Retorna o valor de um campo do registro atual.

Exemplo

`OnFirst`, `OnNext`, `OnEof` e `OnGetValue` são eventos de ponteiro de método (declarados `of object`), portanto exigem métodos nomeados de um formulário ou de uma classe, não procedimentos anônimos. Este exemplo usa dois campos, `FIndex` e `FRows`, declarados no formulário que os contém, junto com os quatro métodos handler abaixo.

```
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` é declarado como `array [0..1] of TArray<string>`. Este padrão funciona em todas as versões do Delphi suportadas, do Delphi 7 ao Delphi 13, já que atribui ponteiros de método comuns, e não métodos anônimos.

TsgcReportPDFExport

Unidade: sgcReports_Export_PDF
Renderiza páginas preparadas através do sgcPDF: fontes TrueType incorporadas e em subconjunto, fontes base 14, texto da direita para a esquerda, rotação, recorte, texto justificado, formas, imagens e códigos de barras vetoriais. Produz saída marcada (tagged) PDF/UA-1 por padrão e saída arquivística PDF/A-3 opcional. Baseado em `TComponent`.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Compress	Boolean	True	Object streams e uma tabela de referência cruzada compactada; os content streams são sempre compactados.
PDFA	TsgcReportPDFAMode	rpaNone	rpaNone, rpaA3b ou rpaA3a. Veja PDF/A.
PDFUA	Boolean	True	Grava a árvore de estrutura PDF/UA-1; combinável com PDFA. Veja Acessibilidade.
Title / Author / Subject / Keywords	string	—	Metadados do documento PDF.
Language	string	—	Idioma natural BCP 47, gravado pela exportação com tagging.
Creator / Producer	string	—	Metadados do documento PDF.
OnAfterRender	TsgcReportPDFDocumentEvent	—	Disparado assim que cada página é renderizada, antes que o documento seja salvo.
FontManager	TsgcReportFontManager	—	Não published (public). Veja Fontes.
FixedCreationDate	TDateTime	0	Não published (public). Fixa as datas de criação e modificação do PDF/A para uma saída byte a byte idêntica.
FixedDocumentID	string	—	Não published (public). Fixa o /ID ao MD5 deste valor.

Métodos

Método	Descrição
SaveToStream(aPages: TsgcReportPreparedPages; aStream: TStream)	Renderiza as páginas preparadas em um stream.
SaveToFile(aPages: TsgcReportPreparedPages; const aFileName: string)	Renderiza as páginas preparadas em um arquivo.

Exemplo: PDF acessível e arquivístico

```
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

Unidade: sgcReports_Export_XLSX
Exporta as páginas preparadas como uma planilha de uma pasta de trabalho .xlsx. Descende de `TsgcReportGridExport`, que transforma as páginas preparadas em uma grade (uma linha por instância de banda, colunas mescladas a partir das posições dos itens dentro de uma tolerância) compartilhada com TsgcReportCSVExport. O pacote é um ZIP com entradas STORED (Content_Types, relationships, workbook, uma planilha e styles) usando strings inline, gravado sem nenhuma biblioteca zip externa.

Propriedades (próprias)

Propriedade	Tipo	Descrição
SheetName	string	Nome da planilha.
FixedDateTime	TDate-Time	Data e hora de modificação armazenadas nas entradas do ZIP; 0 (padrão) usa a hora atual. Defina-a para uma saída byte a byte idêntica.

Propriedades (herdadas de TsgcReportGridExport)

Propriedade	Tipo	Padrão	Descrição
ExcludePage-Bands	Boolean	True	Exclui da grade as bandas de cabeçalho de página, rodapé de página, cabeçalho de coluna e rodapé de coluna.
Include-BandTypes	TsgcReport-BandTypes	rbtReportTitle..rbtChild	Quais tipos de banda contribuem com linhas.
PageFrom / PageTo	Integer	0	Intervalo de páginas baseado em 1; 0 significa todas as páginas.
DetectNum-bers	Boolean	True	Células de texto que podem ser interpretadas como número são gravadas como células numéricas.
ColumnTole-rance	Double	—	Posições Left mais próximas do que este valor (em pontos) são tratadas como a mesma coluna.
DecimalSepa-rator / Thou-sandSepara-tor	Char	locale	Não published (public). Usado por DetectNum-bers.

Métodos (herdados)

Método	Descrição
SaveToStream(aPages, aStream)	Grava a pasta de trabalho em um stream.
SaveToFile(aPages, const aFileNa-me)	Grava a pasta de trabalho em um arquivo.

BuildGrid(aPages): TsgcReportGrid	Cria a grade intermediária; o chamador assume a posse do resultado.

Exemplo

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

TsgcReportCSVExport

Unidade: sgcReports_Export_CSV
Exporta as páginas preparadas como texto CSV, usando a mesma grade baseada em layout que TsgcReportXLSXExport (veja as propriedades de `TsgcReportGridExport` naquela página: ExcludePageBands, IncludeBandTypes, PageFrom, PageTo, DetectNumbers, ColumnTolerance).

Propriedades (próprias)

Propriedade	Tipo	Padrão	Descrição
Delimiter	Char	','	Delimitador de campo.
QuoteChar	Char	'"'	Caractere de citação; #0 desativa a citação.
LineBreak	string	—	Separador de linha.
WriteBOM	Boolean	True	Grava uma marca de ordem de byte (BOM) UTF-8 no início do arquivo.

Métodos

Método	Retorno	Descrição
GetCSV(aPages: TsgcReportPreparedPages)	string	Cria o texto CSV em memória.
SaveToStream(aPages, aStream)	—	Grava o CSV em um stream.
SaveToFile(aPages, const aFileName)	—	Herdado de TsgcReportGridExport; grava o CSV em um arquivo.

Exemplo

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

TsgcReportImageExport

Unidade: sgcReports_Export_Image
Exporta uma imagem PNG por página, no DPI escolhido, através do renderizador raster do sgcPDF, em todas as plataformas. Baseado em `TComponent`.

Propriedades

Propriedade	Tipo	Padrão	Descrição
DPI	Integer	150	Pixels por polegada das imagens.
PageFrom / PageTo	Integer	0	Intervalo de páginas baseado em 1 para SaveToFiles; 0 significa a primeira / última página.
BackgroundColor	TsgcReportColor	sgcReportColorWhite	Cor de preenchimento antes da renderização; sgcReportColorNone significa branco.
FontManager	TsgcReportFontManager	—	Não published (public). Resolve as faces para itens sem uma face explícita.

Métodos

Método	Retorno	Descrição
PageCount(aPages: TsgcReportPreparedPages)	Integer	Número de páginas que SaveToFiles gravaria, com PageFrom / PageTo aplicados.
SaveToStream(aPages, aPageIndex: Integer, aStream: TStream)	—	Grava uma página (base 0, PageFrom / PageTo ignorados) como PNG.
SaveToFiles(aPages, const aFileNamePattern: string)	—	Grava um arquivo PNG por página do intervalo; aFileNamePattern contém %d, substituído pelo número de página baseado em 1.
ClearCache	—	Descarta o PDF em cache usado internamente para rasterizar; a próxima exportação o regenera.

Exemplo

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

TsgcReportPreviewVCL

Unidade: sgcReports_Preview_VCL
Painel de preview VCL com navegação, zoom, impressão e exportação para PDF. Descende de `TCustomPanel` e hospeda internamente um `TsgcPDFViewerControl`, de modo que um relatório é sempre renderizado em PDF para o preview.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Engine	TsgcReportEngine	—	O engine que Prepare executa.
ShowToolbar	Boolean	True	Mostra a barra de ferramentas interna de navegação e zoom.
OnPageChange	TNotifyEvent	—	Disparado quando a página exibida muda.
PageCount	Integer	—	Não published (public), somente leitura.
CurrentPage	Integer	—	Não published (public). Baseado em 1, 0 quando nada é exibido.
Zoom	Double	—	Não published (public). Percentual, 100 é o tamanho real.
FontManager	TsgcReportFontManager	—	Não published (public). Veja LoadPrepared.
Viewer	TsgcPDFViewerControl	—	Não published (public), somente leitura. O viewer sgcPDF subjacente.

Métodos

Método	Descrição
Prepare	Executa Engine.Prepare e mostra o resultado.
LoadPrepared(aPages: TsgcReportPreparedPages)	Mostra páginas preparadas em outro lugar, renderizadas com FontManager, ou Engine.FontManager, ou um manager interno com as fontes do sistema; informe o manager com o qual as páginas foram preparadas.
Clear	Limpa o preview.
FirstPage / PriorPage / NextPage / LastPage	Navegação.
ZoomIn / ZoomOut / FitWidth / FitPage	Controle de zoom.
Print(const aTitle: string = '')	Imprime o documento exibido.
SaveToPDF(const aFileName: string)	Grava o PDF exibido pelo preview.

Exemplo

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

TsgcReportPreviewFMX

Unidade: sgcReports_Preview_FMX
Preview FireMonkey para desktop e dispositivos móveis, espelhando TsgcReportPreviewVCL sobre um `TControl` e um `TsgcPDFViewerFMX`. Executa em Windows, Android e iOS.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Engine	TsgcReportEngine	—	O engine que Prepare executa.
ShowToolbar	Boolean	True	Mostra a barra de ferramentas interna de navegação e zoom.
OnPageChange	TNotifyEvent	—	Disparado quando a página exibida muda.
PageCount	Integer	—	Não published (public), somente leitura.
CurrentPage	Integer	—	Não published (public). Baseado em 1.
Zoom	Double	—	Não published (public). Percentual, 100 é o tamanho real.
Viewer	TsgcPDFViewerFMX	—	Não published (public), somente leitura.

Os métodos e o restante da superfície (Prepare, LoadPrepared, navegação, zoom, SaveToPDF) são iguais aos de TsgcReportPreviewVCL. Veja Como fazer: um relatório FMX móvel para um exemplo completo, incluindo a implantação de fontes em iOS e Android.

TsgcReportHTTPEndpoint

Unidade: sgcReports_HTTP (pacote sgcReportsHTML)
Serve relatórios registrados através de um servidor HTTP sgcWebSockets: páginas PDF, XLSX, CSV, PNG e um JSON de informações, com parâmetros de consulta em lista branca e um cache de páginas preparadas. Requer sgcWebSockets edição Professional ou superior. Baseado em `TComponent`.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Server	TsgcHTTPServer_Base	—	O servidor HTTP a ser conectado. Opcional: HandleRequest pode ser chamado, em vez disso, a partir do seu próprio OnCommandGet.
BasePath	string	—	Prefixo de URL que o endpoint possui, por exemplo /reports.
CacheSize	Integer	8	Conjuntos de parâmetros preparados mantidos por relatório; 0 desativa o cache.
MaxParamLength	Integer	256	Maior comprimento aceito para o valor de um parâmetro de consulta.
DefaultDPI / MaxDPI	Integer	96 / 300	DPI usado e permitido para a rota de página PNG.
OnBeforePrepare	TsgcReportHTTPBeforePrepareEvent	—	Disparado antes de um relatório ser preparado para uma requisição; pode validar ou rejeitar parâmetros (por exemplo, responder 403).
DesignerEnabled	Boolean	False	Habilita as rotas do designer (schema, definition, preview) usadas por TsgcReportHTMLDesigner.
MaxDefinitionSize	Integer	1048576	Maior corpo de definição de relatório aceito, em bytes.
OnBeforeSaveDefinition	TsgcReportHTTPBeforeSaveDefinitionEvent	—	Disparado antes que uma definição enviada pelo designer seja salva.

Métodos

Método	Retorno	Descrição
RegisterReport(const aName: string; aEngine: TsgcReportEngine)	—	Registra um engine sob um nome; o engine não tem sua posse assumida. aName: letras, dígitos, '_' e '-'.
UnregisterReport(const aName: string)	—	Remove o registro de um relatório.

FindEngine(const aName: string)	TsgcReportEngine	Procura um engine registrado.
GetReportParameters(const aName, aList: TStrings)	Boolean	Copia as Variables (Name=Value) do engine de um relatório; False se desconhecido.
ClearCache(const aName: string = '')	—	Descarta as páginas preparadas em cache de um relatório, ou de todos os relatórios quando aName está vazio.
GetReportPath(const aName: string)	string	Caminho de URL de um relatório, sem extensão, por exemplo /reports/sales.
HandleRequest(ARequestInfo, AResponseInfo)	Boolean	Atende a uma requisição que pertence a BasePath e retorna True; chame a partir do seu próprio OnCommandGet quando Server não estiver atribuído.

Rotas

Para um relatório registrado como "sales" sob BasePath "/reports": `/reports/sales.pdf`, `/reports/sales.xlsx`, `/reports/sales.csv`, `/reports/sales/page/1.png?dpi=96` e `/reports/sales/info` (JSON: name, title, page count, parameters). Somente os parâmetros de consulta cujo nome existe nas Variables do engine chegam ao relatório.

Rotas do designer

Disponíveis apenas quando `DesignerEnabled` é `True` (caso contrário toda rota responde 404), e usadas por TsgcReportHTMLDesigner:

Método	Rota	Descrição
GET	BasePath/schema.json	JSON Schema das definições de relatório, usado para alimentar o inspetor de propriedades do designer.
GET	BasePath/{name}/definition	A definição de relatório atual, como JSON.
PUT or POST	BasePath/{name}/definition	Valida o corpo, dispara `OnBeforeSaveDefinition`, e substitui o relatório do engine registrado.
POST	BasePath/{name}/preview.png?page=1&dpi=96	Renderiza a definição no corpo da requisição sem substituir o relatório armazenado; a resposta traz o cabeçalho `X-Report-Page-Count`.

O corpo da requisição das rotas de definição e preview deve ser `application/json` e não maior que `MaxDefinitionSize`; uma requisição que falhe em qualquer uma dessas verificações, ou que falhe na validação do schema, é rejeitada com um corpo de erro JSON (`ok`, `path`, `message`), onde `path` é o caminho JSON do membro problemático. Consulte Como fazer: servir relatórios via HTTP para um servidor completo, e Como fazer: o designer de relatórios web para o designer em si.

TsgcReportHTMLViewer

Unidade: sgcReports_HTML_Viewer (pacote sgcReportsHTML)
Visualizador em navegador para um relatório servido por um TsgcReportHTTPEndpoint: navegação entre páginas, um formulário de parâmetros e links de download, incorporado em uma página sgcHTML. Descende de `TsgcHTML-Component`.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Endpoint	TsgcReportHTTPEndpoint	—	O endpoint do qual o viewer lê as rotas do relatório.
ReportName	string	—	Nome do relatório a mostrar, conforme registrado no endpoint.
DPI	Integer	96	DPI usado para as imagens de página.
ShowParameters	Boolean	—	Mostra um formulário de entrada para os parâmetros do relatório (Variables do engine).
ShowToolbar	Boolean	—	Mostra a navegação de páginas e os links de download.

Exemplo

```
reportViewer.Endpoint := reportEndpoint;
reportViewer.ReportName := 'sales';
reportViewer.ShowParameters := True;
```

Consulte Como fazer: servir relatórios via HTTP, que hospeda o viewer em "/" ao lado do endpoint em "/reports".

TsgcReportHTMLDesigner

Unidade: sgcReports_HTML_Designer (pacote sgcReportsHTML)
Designer de bandas baseado em navegador que edita uma definição de relatório em JSON de acordo com o schema, mostrando uma pré-visualização ao vivo através do mesmo endpoint. Descende de `TsgcHTMLComponent`. Requer `TsgcReportHTTPEndpoint.DesignerEnabled := True`. Consulte Como fazer: o designer de relatórios web.
A página que ele renderiza é um designer de bandas completo, com arrastar e soltar, não um simples editor de formulário:

- Uma árvore de páginas, bandas e itens à esquerda, orientada pela definição atual.
- Uma tela de banda, em milímetros, independentemente do `Units` do próprio relatório, onde os itens são arrastados e redimensionados com o ponteiro, ajustam-se à grade e podem ser ajustados finamente com o teclado.
- Um inspetor de propriedades gerado a partir do `schema.json`, de modo que ele segue os campos e enumerações da definição do relatório em vez de um formulário fixo.
- Save, que envia a definição para `{name}/definition` e destaca o campo problemático quando o servidor a rejeita, a partir do caminho JSON na resposta de erro.
- Preview, que envia a definição para `{name}/preview.png` e mostra a página renderizada sem salvar.
- Undo e redo sobre o histórico de edição em memória, e um botão Download JSON que salva a definição atual como um arquivo.

Propriedades

Propriedade	Tipo	Descrição
Endpoint	TsgcReportHTTPEndpoint	O endpoint do qual o designer lê as rotas de schema, definição e preview; deve ter `DesignerEnabled` definido como `True`.
ReportName	string	Nome da definição de relatório sendo editada.
DesignerHeight	Integer	Altura, em pixels CSS, do designer incorporado, padrão 640.

Exemplo

```
reportEndpoint.DesignerEnabled := True;
reportDesigner.Endpoint := reportEndpoint;
reportDesigner.ReportName := 'sales';
```

TsgcReportBuilder

Unidade: sgcReports_Fluent
API fluente code-first: estilos, páginas, bandas e itens construídos com chamadas encadeadas que retornam `Self` ou um sub-builder com escopo próprio, de modo que um relatório inteiro seja uma única árvore de expressão. Funciona no Delphi 7 (uma classe simples, sem necessidade de métodos anônimos). Constrói e possui seu próprio `TsgcReport` e `TsgcReportEngine`, a menos que um relatório existente seja passado ao construtor.

Construtores

Construtor	Descrição
Create	Cria e possui um novo TsgcReport.
Create(aReport: TsgcReport)	Constrói dentro de um relatório existente, sem assumir a posse dele.

Métodos

Método	Retorno	Descrição
Title / Author / Subject / Keywords / Language(const aValue: string)	TsgcReportBuilder	Define a propriedade correspondente de TsgcReport.
Units(aUnits: TsgcReportUnits)	TsgcReportBuilder	Define TsgcReport.Units.
Style(const aName: string)	TsgcReportStyleBuilder	Encontra ou adiciona o estilo aName; encadeie Font, Bold, Italic, Underline, Color, BackColor, Frame, Padding no resultado.
Page(const aName: string = '')	TsgcReportPageBuilder	Adiciona uma página a cada chamada; encadeie chamadas de PaperSize, Margins, Band e de itens Text/Barcode/Image/Line.
Data(aSource: TsgcReportDataSource)	TsgcReportBuilder	Adiciona uma fonte de dados ao engine; sem assumir a posse dela.
Variable(const aName, aValue: string)	TsgcReportBuilder	Define uma variável do engine.
FontManager(aManager: TsgcReportFontManager)	TsgcReportBuilder	Atribui o font manager do engine; sem assumir a posse dele.
Flush	—	Grava os modificadores pendentes da banda ou item atual.
Prepare	TsgcReportPreparedPages	Prepara o relatório; o chamador assume a posse do resultado.
Prepare(aPages: TsgcReportPreparedPages)	—	Sobrecarga que prepara dentro de um objeto de páginas existente.
SaveToPDF(const aFileName) / SaveToPDFStream(aStream)	—	Prepara e exporta através de PDFExport; Title, Author, Subject, Keywords e Language do relató-

		rio são copiados para PDFExport antes de cada gravação.
SaveToXLSX / SaveToXLSXStream	—	Prepara e exporta para XLSX.
SaveToCSV / SaveToCSVStream	—	Prepara e exporta para CSV.
SaveToPNG(const aFileNamePattern: string; aDPI: Integer = 150)	—	Um arquivo PNG por página; aFileNamePattern contém %d.

Propriedades

Propriedade	Tipo	Descrição
Report	TsgcReport	O relatório, com os modificadores pendentes já gravados.
Engine	TsgcReportEngine	O engine do builder, conectável a um TsgcReportPreviewVCL / TsgcReportPreviewFMX.
PDFExport	TsgcReportPDFExport	Criado no primeiro uso, com posse do builder; defina PDFA, PDFUA ou Compress antes de salvar.

Exemplo

Consulte Primeiros passos para um exemplo mínimo, e Como fazer: um relatório agrupado com totais para um relatório completo construído inteiramente com a API fluente.

Definições de Relatório em JSON

Unidade: sgcReports_Definition
Carrega e salva um TsgcReport completo como JSON, de modo que um relatório criado uma vez com a API de modelo ou a API fluente possa ser editado, armazenado e reimplantado sem a necessidade de recompilar.

Envelope

```
{"format": "sgcReports", "version": 1, "report": { ... }}
```

Um membro de string opcional `"$schema"` é aceito no envelope.

Convenções

- Cada propriedade publicada do modelo é gravada com seu nome Delphi em lowerCamelCase (`DataSourceName` torna-se `dataSourceName`, `HAlign` torna-se `hAlign`); o `Name` do componente do relatório não faz parte da definição.
- As enumerações são gravadas como seus identificadores Delphi (`"rbtMasterData"`), os conjuntos como matrizes de identificadores, os booleanos como `true` / `false`, os inteiros e números de ponto flutuante como números JSON; as cores são inteiros `$00RRGGBB`.
- Os subobjetos `TPersistent` são objetos aninhados, as collections são matrizes de objetos de item, e os `TStrings` são matrizes de strings.
- Valores iguais ao seu padrão são omitidos, exatamente como em um DFM.
- Os bytes da imagem incorporada de um item de imagem são gravados como a string base64 `pictureData` de seu objeto `image`.

O carregamento limpa o relatório primeiro. Qualquer erro gera `ESgcReportDefinitionError` com o caminho JSON do membro problemático, por exemplo `report.pages[0].bands[2].height`. Um membro desconhecido é um erro, a menos que `IgnoreUnknown` seja `True`. Streams e arquivos são UTF-8 sem BOM ao salvar; um BOM é tolerado ao carregar.

Funções

Função	Descrição
sgcReportSaveToJSON(aReport: TsgcReport; aIndent: Boolean = True): string	Serializa um relatório em uma string JSON.
sgcReportLoadFromJSON(aReport: TsgcReport; const aJSON: string)	Carrega um relatório a partir de uma string JSON.
sgcReportSaveToJSONFile / sgcReportLoadFromJSONFile	Variantes para arquivo.
sgcReportSaveToJSONStream / sgcReportLoadFromJSONStream	Variantes para stream.
sgcReportGenerateSchema: string	JSON Schema (draft 2020-12) da definição, gerado a partir de RTTI; usado como sgcReports.schema.json.

TsgcReportDefinition

A classe por trás das funções simples, para chamadores que precisam de controle sobre `IgnoreUnknown` ou `Indent`: `SaveToJSON`, `LoadFromJSON`, `SaveToStream` / `LoadFromStream`, `SaveToFile` / `LoadFromFile`, e as propriedades `IgnoreUnknown` (padrão False) e `Indent` (padrão True, indentação de dois espaços e quebras de linha LF).

Exemplo

Consulte Primeiros passos e Como fazer: o designer de relatórios web, que edita um definition.json e um data.json.

Como Fazer: Fatura com Dados Mestre-Detalhe

Este tópico percorre a demo `ReportInvoice` (`delphi\Demos\Delphi\ReportInvoice`): dois `TClientDataSet` (faturas e linhas de fatura) lidos através de dois TsgcReportDataSetSource, vinculados por `MasterSource`, impressos como uma página por fatura, com um cabeçalho de coluna repetido, subtotal, IVA e total, um código QR com o número da fatura, e Página X de Y. O relatório é mostrado em um TsgcReportPreviewVCL e exportado como um arquivo PDF/A-3a que também é PDF/UA-1.

Vincular as duas fontes

```
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);
```

Uma página por fatura

O cabeçalho de grupo que inicia cada fatura define `StartNewPage`, e sua `GroupExpression` é o número da fatura, de modo que um novo grupo (e uma nova página) começa toda vez que ele muda:

```
oBand := oPage.Bands.Add;
oBand.BandType := rbtGroupHeader;
oBand.Name := 'InvoiceHeader';
oBand.Height := 110;
oBand.GroupExpression := 'Invoices.Number';
oBand.StartNewPage := True;
```

O item de título dentro dele é marcado como `rtrHeading1` para o PDF/UA, e o valor do código QR é um modelo de texto com o número da fatura, não uma expressão isolada, de modo que o texto literal ao redor seria mantido:

```
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';
```

A tabela de linhas e seu cabeçalho repetido

O cabeçalho de coluna da tabela de linhas é uma segunda banda `rbtGroupHeader` na mesma `GroupExpression`, com `RepeatOnEveryPage := True`, de modo que ele se repete sempre que as linhas de uma fatura longa continuam em uma segunda página. A banda mestre itera as faturas e não imprime nada; a banda de detalhe itera as linhas:

```
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';
```

Totais e IVA

O rodapé de grupo vê os campos da última linha impressa e da própria fatura, de modo que sua alíquota de IVA vem diretamente de `Invoices.VATRate`, mesmo que o último registro iterado tenha sido uma linha:

```
// 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 acessível e arquivístico

```
// 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;
```

O projeto completo, que compila, é `delphi\Demos\Delphi\ReportInvoice\sgcReports_Demo_Invoice.dpr`.

Como Fazer: Um Relatório Agrupado com Totais

Este tópico percorre a demo `ReportFluent` (`delphi\Demos\Delphi\ReportFluent`): uma lista de vendas lida de `sales.json`, agrupada por região, com um subtotal por região e um total geral, construída inteiramente com TsgcReportBuilder. Um equivalente com fonte JSON e API de modelo, agrupado por região e cliente, é a demo `ReportGroups`.

Fonte de dados e 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 e legendas de coluna

```
  .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 região, uma linha por venda, 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` aceita tanto uma expressão simples (`'Sales.Region'`) quanto uma envolta em `[ ]`. O agregado `Sum` em um rodapé de grupo reinicia a cada região; a mesma expressão na banda de resumo abaixo é executada para o relatório inteiro.

Total geral e 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);
```

Preview e exportação

```
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);
```

O projeto completo, que compila, é `delphi\Demos\Delphi\ReportFluent\sgcReports_Demo_Fluent.dpr`; o mesmo padrão de agrupamento e totais com uma fonte `TDataSet` ou de API de modelo também é mostrado por `ReportGroups`.

Como Fazer: Exportar para XLSX, CSV e PNG

Este tópico percorre a demo `ReportExports` (`delphi\Demos\Delphi\ReportExports`): um relatório é preparado uma vez e depois exportado através de quatro componentes diferentes. As páginas preparadas (`TsgcReportPreparedPages`) são reutilizadas em cada export, de modo que o layout e a travessia dos dados são executados apenas uma vez.

```
sgcEngine.Report := sgcReport;
sgcEngine.AddDataSource(srcOrders);
FPages := sgcEngine.Prepare;
sgcPreview.LoadPrepared(FPages);
```

PDF

```
pdfExport.SaveToFile(FPages, OutputFolder + 'orders.pdf');
```

XLSX

TsgcReportXLSXExport transforma as páginas preparadas em uma grade (uma linha de planilha por instância de banda) antes de gravar a pasta de trabalho:

```
xlsxExport.SheetName := 'Orders';
xlsxExport.SaveToFile(FPages, OutputFolder + 'orders.xlsx');
```

CSV

```
csvExport.Delimiter := ',';
csvExport.SaveToFile(FPages, OutputFolder + 'orders.csv');
```

PNG, um arquivo por página

```
imageExport.DPI := 150;
imageExport.SaveToFiles(FPages, OutputFolder + 'orders_%d.png');
```

Notas

- Por padrão (`ExcludePageBands = True`) os cabeçalhos e rodapés de página e de coluna não se tornam linhas de grade na exportação XLSX e CSV; apenas o título do relatório e as bandas de grupo e de dados o fazem.
- `DetectNumbers` (padrão `True`) grava uma célula de texto que pode ser interpretada como número como uma célula numérica do XLSX, em vez de texto.
- `SaveToFiles` nomeia os arquivos substituindo o número de página baseado em 1 por `%d` no padrão, de modo que `orders_%d.png` produz `orders_1.png`, `orders_2.png`, e assim por diante.

O projeto completo, que compila, é `delphi\Demos\Delphi\ReportExports\sgcReports_Demo_Exports.dpr`.

Como Fazer: Um PDF Acessível Validado com o veraPDF

Este tópico percorre a demo `ReportPDFUA` (`delphi\Demos\Delphi\ReportPDFUA`): um título H1, títulos de grupo H2, uma tabela com linhas de cabeçalho, uma imagem com texto alternativo e um código de barras, exportados como PDF/UA-1, depois como PDF/A-3b, depois como PDF/A-3a e PDF/UA-1 juntos. Veja Acessibilidade e PDF/A para as regras de tagging e os níveis de conformidade.

Marcar o conteúdo

```
// 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;
```

Uma tabela construída a partir de uma banda de cabeçalho de grupo ou de coluna, seguida de linhas de detalhe, é marcada automaticamente; nada mais é necessário para que suas células de cabeçalho sejam marcadas como `rtrTableHeader`.

Exportar as três 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 com o veraPDF

veraPDF é um validador de PDF/A e PDF/UA de código aberto, respaldado pela ISO, executado separadamente do sgcReports como uma ferramenta de linha de comando. Verifique cada arquivo em relação ao perfil correspondente:

```
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
```

Uma execução bem-sucedida relata `compliant: true` para o perfil solicitado; uma falha lista a cláusula ISO e o objeto que a viola, o que é a maneira mais rápida de encontrar um `AltText` ausente ou um item que deveria ter sido marcado como `rtrArtifact`.
O projeto completo, que compila, é `delphi\Demos\Delphi\ReportPDFUA\sgcReports_Demo_PDFUA.dpr`.

Como Fazer: Servir Relatórios via HTTP

Este tópico percorre a demo `ReportHTTPServer` (`delphi\Demos\Delphi\ReportHTTPServer`): um `TsgcWebSocketHTTPServer` na porta 8080 com um TsgcReportHTTPEndpoint servindo um relatório "sales", e uma página sgcHTML em `/` hospedando um TsgcReportHTMLViewer. Necessita do pacote `sgcReportsHTML`: sgcWebSockets (edição Professional ou superior), sgcHTML e sgcReports.

Registrar o relatório e conectar o servidor

```
reportEndpoint.BasePath := '/reports';
reportEndpoint.Server := httpServer;
reportEndpoint.RegisterReport('sales', salesEngine);

httpServer.Port := 8080;
httpServer.Active := True;
```

Rotas 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 parâmetros

O endpoint aceita somente parâmetros de consulta cujo nome existe nas Variables do engine do relatório, de modo que um relatório com uma variável chamada `Region` aceita apenas `?Region=...`. `OnBeforePrepare` pode validar o valor mais a fundo, por exemplo rejeitando qualquer coisa fora de uma lista permitida antes de o relatório ser preparado:

```
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 a página do viewer

```
reportViewer.Endpoint := reportEndpoint;
reportViewer.ReportName := 'sales';
reportViewer.ShowParameters := True;
```

Notas de compilação

Compile com a pasta `Source` do sgcWebSockets no search path. Defina `SGC_CUSTOM_INDY` ao vincular contra os pacotes sgcWebSockets lançados (construídos sobre o fork sgcIndy), ou `SGC_REPORTS_STOCK_INDY` quando o sgcWebSockets foi construído contra o Indy que acompanha a IDE.
O projeto completo, que compila, é `delphi\Demos\Delphi\ReportHTTPServer\sgcReports_Demo_HTTPServer.dpr`.

Como Fazer: Um Daemon de Relatórios em Linux64

Este tópico percorre a demo `ReportLinuxDaemon` (`delphi\Demos\Delphi\ReportLinuxDaemon`), uma aplicação de console que roda sem alterações em Windows e Linux64, já que o núcleo do sgcReports não tem dependência de VCL ou FMX. Veja Segurança em threads e uso em servidor para o padrão de pool que ela usa.

Arquivos de job

O daemon monitora uma pasta de jobs em busca de arquivos de job `*.json`. Um job informa uma definição de relatório, um arquivo de dados e um formato de saída (`pdf`, `xlsx`, `csv` ou `png`); caminhos relativos no job são resolvidos em relação à pasta do próprio arquivo de job. Um job é reivindicado renomeando-o para `.json.working`, a saída é gravada ao lado dele, e o job termina como `.json.done` ou `.json.failed`.

Pool de workers

Um pool de 4 threads de trabalho renderiza jobs concorrentemente. Cada thread carrega a definição de relatório do job com `sgcReportLoadFromJSON` em seu próprio `TsgcReport`, cria seu próprio `TsgcReportEngine` e fonte de dados, prepara, exporta, e então libera tudo o que criou, seguindo o padrão compartilhado e somente leitura descrito em Segurança em threads e uso em servidor.

Executando-o

```
sgcReports_Demo_LinuxDaemon JOBS_FOLDER
sgcReports_Demo_LinuxDaemon --sample FOLDER   -- writes sample.job plus its report definition and data
```

SIGTERM ou SIGINT no Linux, e Ctrl+C no Windows, encerram o daemon de forma limpa: nenhum novo job é reivindicado, os jobs em execução terminam, e o processo é encerrado com o código 0.

Unidade systemd

`sgcreports-daemon.service` executa o executável compilado sob um usuário dedicado, monitora `/var/lib/sgcreports/jobs`, reinicia em caso de falha e para com SIGTERM com um tempo limite de 60 segundos para os jobs em execução terminarem. Os logs vão para stdout, lidos com:

```
journalctl -u sgcreports-daemon -f
```

O projeto completo, que compila, é `delphi\Demos\Delphi\ReportLinuxDaemon\sgcReports_Demo_LinuxDaemon.dpr`.

Como Fazer: Um Relatório FMX Móvel

Este tópico percorre a demo `ReportMobileFMX` (`delphi\Demos\Delphi\ReportMobileFMX`), uma pequena nota de entrega A5 (um código QR com o número da nota, linhas de item, total de unidades e uma linha de assinatura) construída com a API fluente e mostrada em um TsgcReportPreviewFMX. Direcionada a Win32, Win64, Android64 e iOS a partir de um único projeto.

Local de gravação

O botão Save PDF grava `sgcReports_DeliveryNote.pdf` na pasta de documentos do aplicativo e mostra o caminho:

```
uses System.IOUtils;
...
vBuilder.SaveToPDF(TPath.Combine(TPath.GetDocumentsPath,
  'sgcReports_DeliveryNote.pdf'));
```

Fontes em dispositivos móveis

A demo usa as fontes já presentes no sistema através da descoberta de fontes (veja Fontes). O iOS não tem uma pasta de fontes do sistema legível a partir de um aplicativo em sandbox, portanto, quando nenhuma fonte TrueType adequada é encontrada, a demo carrega `DejaVuSans.ttf` de sua própria pasta; `DejaVu-LICENSE.txt` é a licença dessa fonte e é distribuído junto com ela.

Implantando a fonte

Na IDE, Project > Deployment, adicione `DejaVuSans.ttf` para cada plataforma móvel com estes caminhos remotos:

Plataforma	Caminho remoto
Android64	assets\internal\
iOS	StartUp\Documents\

No Windows a fonte é lida da pasta do executável, portanto copie-a para lá quando o sistema não tiver uma fonte adequada instalada.
O projeto completo, que compila, é `delphi\Demos\Delphi\ReportMobileFMX\sgcReports_Demo_MobileFMX.dpr`; requer sgcReports, sgcPDF e FireMonkey.

Como Fazer: O Designer de Relatórios Web

TsgcReportHTMLDesigner é um designer de bandas baseado em navegador que edita uma definição de relatório em JSON de acordo com `sgcReports.schema.json`, com uma pré-visualização ao vivo servida pelo mesmo TsgcReportHTTPEndpoint que um viewer ou uma aplicação usa para servir o relatório finalizado.

Habilitá-lo

```
reportEndpoint.DesignerEnabled := True;
reportDesigner.Endpoint := reportEndpoint;
reportDesigner.ReportName := 'sales';
```

`DesignerEnabled` habilita as rotas de schema, definition e preview que a página do designer chama; deixe-o `False` em um endpoint de produção que serve apenas relatórios finalizados, e habilite-o somente onde a edição é pretendida, protegida por sua própria autenticação.

O que o designer de navegador oferece

A página servida por `TsgcReportHTMLDesigner` é um designer de bandas completo, com arrastar e soltar: uma árvore de páginas, bandas e itens; uma tela de banda em milímetros com arrastar, redimensionar, ajuste à grade e ajuste fino pelo teclado; um inspetor de propriedades gerado a partir de `schema.json`; Save (com o campo problemático destacado a partir do caminho JSON de um erro de validação do lado do servidor) e Preview; Undo e redo; e um botão Download JSON. Veja TsgcReportHTMLDesigner para a lista completa de propriedades, e TsgcReportHTTPEndpoint para as rotas do designer que ele chama (`schema.json`, `{name}/definition` e `{name}/preview.png`).

Editando uma definição sem o designer de navegador

A demo `ReportJSONDefinition` (`delphi\Demos\Delphi\ReportJSONDefinition`) mostra o mesmo ciclo completo a partir de código Delphi, útil para um editor desktop ou para testes automatizados de uma definição: ela edita `definition.json` e seu `data.json` em dois memos, um botão Render carrega a definição com `sgcReportLoadFromJSON` e mostra o preview, os erros são relatados na barra de status junto com o caminho JSON do membro problemático (veja Definições de relatório em JSON), e um botão Save definition grava o relatório atual de volta em JSON com `sgcReportSaveToJSON`.

O projeto completo, que compila, para o editor baseado em código é `delphi\Demos\Delphi\ReportJSONDefinition\sgcReports_Demo_JSONDefinition.dpr`.

Introdução

sgcCharts é uma biblioteca de componentes Delphi para gráficos de dados ao vivo. Uma fonte de dados adiciona pontos ou candles OHLC a um buffer circular, e uma tela desenha um snapshot decimado desse buffer, de modo que um feed que publica milhares de atualizações por segundo ainda renderiza a uma taxa de quadros estável.
A biblioteca visa VCL, FireMonkey e renderização sem interface visual (saída PNG ou SVG sem janela), de modo que o mesmo código de buffer e de fonte de dados desenha em um formulário, em uma tela móvel ou dentro de uma aplicação de console ou serviço.
Esta ajuda cobre as partes do sgcCharts que existem hoje: as fontes de dados que alimentam um buffer (TsgcChartDataSource e suas descendentes), a ponte sgcWebSockets que alimenta um buffer a partir de uma conexão ao vivo (Visão geral da Bridge), e os próprios buffers e funções de decimação (TsgcChartBuffer).

- Visão Geral: o modelo, as telas, os buffers e as fontes de dados, e como se encaixam.
- Edições: o que Registered, Trial e Community incluem cada uma.
- Início Rápido: estado atual do caminho de introdução.
- Fontes de Dados: manual, dataset, JSON e CSV.
- Bridge: MQTT, WebSocket, SSE e Binance, por meio do sgcWebSockets.
- Referência: as classes de buffer e as funções de decimação.

Arquitetura

O sgcCharts separa os dados que um gráfico exibe da superfície em que são desenhados, de modo que os mesmos dados podem alimentar um formulário VCL, uma visão FireMonkey ou uma renderização PNG/SVG sem interface visual.

Um modelo, várias telas

Os dados de um gráfico residem em um buffer, não em um controle. **TsgcChartBuffer** mantém pontos X/Y e **TsgcChartOHLCBuffer** mantém candles OHLC (abertura, máxima, mínima, fechamento, volume). Ambas são classes simples sem dependência de VCL ou FireMonkey, portanto podem ser criadas, alimentadas e lidas a partir de uma aplicação de console, um serviço ou uma thread em segundo plano. O desenho é uma etapa separada: uma tela lê um snapshot decimado de um buffer e o desenha, usando uma tela rasterizada (`sgcUICore_Canvas_Raster`) ou uma tela SVG (`sgcUICore_Canvas_SVG`) por trás da superfície compartilhada `sgcUICore_Canvas`.
A camada de canvas, raster, fonte TrueType, SVG e temas que desenha estes gráficos não faz parte do sgcCharts em si. Ela é o pacote compartilhado **sgcUICore** (units Delphi `sgcUICore_*` no pacote de runtime sgcUICore, pacotes .NET `esegece.sgcUICore` e `esegece.sgcUICore.Rendering`). Ele é instalado automaticamente com o sgcCharts e compartilhado com os outros produtos visuais da eSeGeCe.

Buffers circulares com decimação min/max

Ambas as classes de buffer são buffers circulares de capacidade fixa. No modo `cbmOverwrite`, um buffer cheio descarta seu ponto mais antigo na próxima adição; no modo `cbmGrow`, sua capacidade dobra em vez disso. Todo método público de um buffer adquire um bloqueio interno, de modo que uma thread de feed pode adicionar dados enquanto uma thread de renderização lê um snapshot sem que as duas colidam. **Snapshot** e **SnapshotRange** copiam os dados que o chamador precisa para um registro de propriedade do chamador sob esse bloqueio e o liberam imediatamente; o chamador então decima e desenha a partir da cópia, fora do bloqueio.
`sgcChartDecimate` reduz um snapshot a aproximadamente um conjunto de pontos por coluna de pixel, usando `dmMinMax` (o primeiro, o mínimo, o máximo e o último ponto de cada coluna, o que mantém uma polilinha ao vivo visualmente exata) ou `dmLTTB` (Largest Triangle Three Buckets, para uma série estática). `sgcChartDecimateOHLC` mescla os candles que caem na mesma coluna de pixel da mesma forma. Veja TsgcChartBuffer e Decimação.

Gráfico, séries e eixos

TsgcChart é o modelo de gráfico: um componente não visual que mantém os eixos, as séries, o título, a legenda, as margens, o tema e a paleta, mas não desenha nada por si só. Uma série é, ela mesma, um componente, não uma parte aninhada do gráfico; atribuir sua propriedade **Chart** é o que a registra, e uma série lê seu próprio buffer diretamente (por meio de um de seus tipos concretos de série) em vez de passar pelo gráfico. Uma visão chama **TsgcChart.Render** para desenhar todo o modelo em uma TsgcUICanvas, de modo que o mesmo gráfico, com as mesmas séries e eixos, pode ser exibido por mais de uma visão, ou renderizado sem interface visual para PNG ou SVG sem visão alguma. Veja TsgcChart e TsgcChartSeries.

Fontes de dados

Uma descendente de **TsgcChartDataSource** lê registros de algum lugar (uma aplicação chamando **AppendValue** diretamente, um `TDataSet`, um documento JSON ou texto CSV) e adiciona os pontos ou candles correspondentes aos buffers atribuídos às suas propriedades **Buffer** e **OHLCBuffer**. Onde os campos de um registro residem é descrito uma única vez, em **Mapping**, e toda fonte de dados o lê da mesma forma. Veja Fontes de Dados.

Ponte sgcWebSockets

O pacote opcional `sgcChartsWS` adiciona fontes de dados que leem a partir de uma conexão sgcWebSockets ativa: um tópico MQTT, um stream de mensagens WebSocket, um stream de Server-Sent Events, ou a API pública de dados de mercado da Binance. Cada uma vincula o evento existente do componente cliente ao qual está aponta-

da, de modo que o próprio handler da aplicação continua disparando junto com o gráfico. Este pacote exige o sgcWebSockets; o pacote principal `sgcCharts` não. Veja Bridge.

Edições

Veja Edições para o que Registered, Trial e Community incluem cada uma.

Edições

O sgcCharts é construído a partir do mesmo código-fonte para cada edição; a edição apenas muda quais unidades uma compilação inclui.

Edição	Inclui
Registered	Toda unidade: as fontes de dados principais, os indicadores e painéis financeiros, e as fontes de dados ao vivo (streaming), incluindo a ponte sgcWebSockets. Sem aviso de trial.
Trial	As mesmas unidades que Registered. Uma compilação feita com `SGC_TRIAL` exibe um aviso único assim que a janela de lançamento indicada na compilação tiver passado; nada mais muda.
Community	O núcleo completo de gráficos sem limites funcionais: as oito séries (linha, linha rápida, linha em degraus, área, barra/coluna, dispersão, candlestick e OHLC), as visões VCL, FMX e sem interface visual, os buffers, as fontes de dados principais (manual, dataset, JSON, CSV) e a interação. Uma compilação feita com `SGC_COMMUNITY` exclui o pacote financeiro (painéis, séries de indicador, séries de volume, agregador, ferramentas de desenho, marcador de último valor e faixa de compra/venda), e a edição Community não distribui as unidades e pacotes da ponte sgcWebSockets.

A edição é uma escolha em tempo de compilação em `sgcVer.inc` (`SGC_COMMUNITY`, que também desativa `SGC_TRIAL`), não uma opção em tempo de execução.

Instalação

O sgcCharts faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcCharts** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcCharts

- **sgcUICore**, a base compartilhada do sgcCharts e dos produtos de layout. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcCharts é selecionado.
- **sgcCharts for sgcWebSockets**, listado em **sgcWebSockets bridges**. Instala os pacotes sgcChartsWS com as fontes de dados ao vivo da ponte sgcWebSockets. O instalador só o oferece quando o sgcWebSockets já está instalado na máquina e a sua licença inclui o sgcCharts. Caso contrário, ele aparece esmaecido.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13`.
`Demos\Charts`	Os demos do sgcCharts. Veja Demos.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcCharts.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcCharts, junto com sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard. As pontes sgcWebSockets não fazem parte do instalador Community, e o sgcPDF e o sgcMaps não fazem parte do Community. Veja Edições para saber o que a compilação Community deixa de fora.

Migrando de uma instalação autônoma do sgcCharts

Se o sgcCharts foi instalado antes com o seu próprio instalador autônomo, o instalador do sgcUI detecta essa instalação, junto com a pasta do sgcUICore gravada pelo instalador autônomo. Ele lista os library paths e os pacotes de design dessas instalações na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho. A página só é exibida quando uma instalação antiga é encontrada.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcChartsD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcCharts<V>` / `dclsgcCharts<V>`	Os componentes do sgcCharts (runtime / design time).
`sgcChartsWS<V>` / `dclsgcChartsWS<V>`	A ponte sgcWebSockets (runtime / design time). Requer o sgcWebSockets.

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcCharts`, depois `dclsgcCharts`, e os pacotes da ponte por último. Em seguida, instale os pacotes de design time `dcl*` e adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcChartsC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

.NET

Nada muda para o .NET. O sgcCharts continua sendo distribuído como seus próprios pacotes NuGet (`esegece.sgcCharts` e os pacotes complementares). Veja a visão geral do .NET.

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	TsgcVCLChart no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	TsgcFMXChart a partir do Delphi XE6 (SGC_CHARTS_FMX), em Win32, Win64, macOS, iOS e Android.
Linux64	As unidades independentes de framework, TsgcChart e TsgcChartRenderer com a saída PNG e SVG sem interface, a partir do Delphi 10.2 (SGC_CHARTS_LINUX).
.NET	Um port em C# para WinForms, WPF e Avalonia, com saída PNG e SVG sem interface, veja Visão Geral do .NET.

Pacotes

O sgcCharts traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcChartsD13.dpk para o Delphi 13 e sgcChartsC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcCharts	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, dbrtl, sgcUICore
sgcChartsVCL	Runtime, VCL	sgcCharts_VCL (TsgcVCLChart)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcCharts
sgcChartsFMX	Runtime, FireMonkey, a partir do XE6	sgcCharts_FMX (TsgcFMXChart)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcCharts
dclsgcCharts	Tempo de projeto, VCL	Registra TsgcChart, TsgcChartRenderer e TsgcVCLChart na página **SGC Charts** da paleta.	designide, rtl, vcl, sgcCharts, sgcChartsVCL
dclsgcChartsFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXChart na página **SGC Charts** da paleta.	designide, rtl, fmx, sgcCharts, sgcChartsFMX, dclsgcCharts
sgcChartsWS, dclsgcChartsWS	Runtime e tempo de projeto, ponte do sgcWebSockets	A ponte de dados em tempo real, independente de framework.	rtl, sgcCharts, sgcWebSockets

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcChartsFMX exige o dclsgcCharts, portanto instale primeiro o dclsgcCharts.

Versões do IDE e plataformas de destino

Delphi	sgcCharts	sgc-ChartsVCL	sgcChartsFMX	dclsgcCharts, dclsgcChartsFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcCharts, sgcChartsVCL	sgcChartsFMX	dclsgcCharts, dclsgcChartsFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcCharts)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcCharts)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcChartsC13.cbproj, sgcChartsVCLC13.cbproj, sgcChartsFMXC13.cbproj, dclsgcChartsC13.cbproj e dclsgcChartsFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXChart) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcChartsVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcCharts_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcCharts é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa TsgcChart e TsgcChartRenderer, que renderiza um gráfico em PNG ou SVG sem janela. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Veja Também

- Instalação
- Edições
- Arquitetura
- Solução de Problemas

Início Rápido

O sgcCharts separa o **modelo** do gráfico da **visão** que o desenha. Um TsgcChart mantém os eixos, as séries e os dados; não desenha nada por si só. Uma visão, um controle VCL, um controle FireMonkey ou um renderizador sem interface visual, desenha esse modelo sempre que ele muda. Este passo a passo constrói um gráfico etapa por etapa: primeiro em um formulário VCL, depois alimentado com dados ao vivo, e por fim renderizado sem visão alguma para um serviço ou aplicação de console.

1. O modelo de gráfico e a visão VCL

Crie um **TsgcChart** e dê a ele um eixo inferior e um esquerdo, depois crie um **TsgcVCLChart** e aponte sua propriedade **Chart** para o modelo. A visão se registra como um listener e se repinta sempre que o modelo muda; nada mais é necessário para ver um gráfico (vazio) no formulário.

```
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. Adicionando uma série e dados estáticos

Uma série é um componente por direito próprio. Crie uma TsgcChartSeries_Line, aponte sua propriedade **Chart** para o modelo, nomeie os eixos contra os quais ela é plotada, e chame **AddXY** uma vez por ponto. Adicione as duas linhas abaixo ao construtor acima, depois que a visão for criada.

```
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. Alimentando com dados ao vivo

Para continuar adicionando pontos depois que o formulário for exibido, coloque um `TTimer` e chame **AddY** a partir do seu evento **OnTimer**. **AddY** adiciona um ponto no próximo valor X corrente, o que é conveniente quando X é um índice de amostra ou uma contagem de ticks em vez de um valor que o chamador já possui.

```
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;
```

Um timer chamando **AddY** é adequado para um teste rápido, mas uma aplicação real que exibe um feed ao vivo deve usar um dos componentes de fonte de dados em vez disso: **TsgcChartDataSource_Manual** para valores que a aplicação já tem em mãos, **TsgcChartDataSource_DataSet** para um `TDataSet`, ou **TsgcChartDataSource_JSON** / **TsgcChartDataSource_CSV** para um documento ou um arquivo. Eles acrescentam adições thread-safe, notificações limitadas e (para a fonte de dataset) mapeamento de campos que um timer simples não oferece. Para um feed que já chega por MQTT, um WebSocket, Server-Sent Events ou um stream de mercado da Binance, as fontes de dados de ponte se conectam diretamente ao cliente sgcWebSockets correspondente e não precisam de nenhuma sondagem.
Juntando tudo, os passos 1 a 3 resultam no programa completo e compilável abaixo. O formulário não usa nenhum `.dfm`, ele constrói seu próprio layout em código, motivo pelo qual o construtor chama **CreateNew** em vez do usual **Create** herdado.

```
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. Renderização sem interface visual, sem visão alguma

Um serviço, um módulo ISAPI ou uma aplicação de console não tem formulário onde colocar uma visão, e não precisa de um: o mesmo **TsgcChart** e as séries renderizam por meio de um TsgcChartRenderer em vez de um **TsgcVCLChart**. **TsgcChartRenderer** não tem dependência de VCL ou FMX, portanto este programa não se vincula a nada além das unidades principais do 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.
```

Executá-lo grava `quickstart.png` ao lado do executável, sem janela, sem laço de mensagens e sem necessidade de área de trabalho. Veja TsgcChartRenderer para renderizar para SVG, para um stream, ou para um buffer de pixels bruto em vez disso.

Para onde ir a seguir

- Séries: linha, linha rápida, linha em degraus, área, barra, dispersão e séries candlestick, e o que cada uma desenha.
- Visões: **TsgcVCLChart**, **TsgcFMXChart** e **TsgcChartRenderer**.
- Interação: zoom, pan, mira e dicas de ferramenta em uma visão visual.
- Financeiro: indicadores, osciladores, volume e gráficos financeiros multipainel.
- Fontes de Dados: feeds manual, dataset, JSON e CSV.
- Fontes de Dados de Ponte: feeds ao vivo MQTT, WebSocket, SSE e Binance por meio do sgcWebSockets.
- Demos: os projetos de demonstração executáveis nos quais este início rápido se baseia.

Veja Também

- TsgcChart
- TsgcChartSeries
- TsgcVCLChart
- TsgcChartRenderer
- TsgcChartBuffer

Demos

A distribuição do sgcCharts inclui um projeto de demonstração para cada cenário desta ajuda: um feed ao vivo, múltiplos eixos, um gráfico candlestick a partir de um CSV, um gráfico vinculado a um dataset, temas e HiDPI, um feed ao vivo conectado por meio do sgcWebSockets, um gráfico móvel FireMonkey, e renderização sem interface visual sem visão alguma. Os projetos Delphi ficam sob `Demos\Delphi\<Name>\` e as contrapartes C++Builder sob `Demos\CBuilder\<Name>\`; toda pasta de demonstração traz um `readme.html` com os detalhes.

Demonstrações VCL

Estas demonstrações colocam um **TsgcChart**, um **TsgcVCLChart**, as séries e as fontes de dados em um formulário em tempo de design. Compilam do Delphi 7 ao Delphi 13, Win32 e Win64.

Demonstração	O que mostra	Pasta
LiveLine	Um **TsgcChartSeries_FastLine** com um buffer circular de 1.000.000 de pontos, alimentado por uma thread em segundo plano por meio de um **TsgcChartDataSource_Manual** a uma taxa de 1.000 a 200.000 pontos por segundo, com uma **TimeWindow** de 10 s / 60 s / buffer inteiro, uma alternância de tema claro/escuro, e uma mira e dica de ferramenta sobre um milhão de pontos.	Delphi\LiveLine\
MultiAxis	Três séries de linha em três eixos de valor (esquerdo, direito e um segundo eixo esquerdo), um eixo inferior de data/hora e a legenda à direita; cada série escolhe seus eixos com **AxisX** e **AxisY**, com zoom por roda, zoom por caixa, pan e reset em cada eixo.	Delphi\MultiAxis\
Candlestick	Um ano de dados diários OHLCV de um CSV compilado no executável, carregado por **TsgcChartDataSource_CSV** em uma série candlestick com o volume como barras em um segundo eixo; os nomes das colunas são mapeados para X, abertura, máxima, mínima, fechamento e volume com datas ISO 8601, e um botão alterna entre barras candlestick ou OHLC.	Delphi\Candlestick\
DataSetChart	Um `TClientDataSet` com 1.000 linhas, exibido em um `TDBGrid` e lido por dois componentes **TsgcChartDataSource_DataSet** em uma série de linha e uma série de barras; **XField** e **YField** nomeiam os campos do dataset, e editar uma linha na grade e recarregar atualiza o gráfico.	Delphi\DataSetChart\
Themes	O mesmo gráfico nos temas claro e escuro lado a lado, com um seletor de escala (1.0, 1.5, 2.0) e exportação para PNG e SVG por meio de **TsgcVCLChart.SaveToPNG** e **SaveToSVG**. O projeto é `Themes\ChartThemes.dpr` .	Delphi\Themes\
PieDonut	Uma **TsgcChartSeries_Pie** e uma rosca lado a lado, alimentadas ao vivo a partir de uma pequena `IsgcUITable` editada em uma grade de strings por meio de **TsgcChartDataSource_Table**, com um controle deslizante do limite de Others, explosão ao clicar, uma chave de modo de rótulo e exportação para PNG e SVG.	Delphi\PieDonut\
RadarCompare	Três produtos em oito critérios com **TsgcChartSeries_Radar**: preenchido ou contorno, grade poligonal ou circular, e um temporizador que move as pontuações.	Delphi\RadarCompare\

HeatmapLive	Um espectrograma **TsgcChartSeries_Heatmap** que acrescenta 30 linhas por segundo em um eixo de data e hora com **Capacity** 300, e uma matriz de região por mês carregada de uma `IsgcUITable` com eixos de categoria, rótulos de valor e uma escala de cores contínua ou em degraus.	Delphi\HeatmapLive\
TreemapFunnel	Um **TsgcChartSeries_Treemap** de três níveis com detalhamento e breadcrumb, e um **TsgcChartSeries_Funnel** ao vivo de vendas com seis etapas.	Delphi\TreemapFunnel\

Demonstrações da Bridge sgcWebSockets

Estas demonstrações alimentam os gráficos com dados ao vivo de componentes sgcWebSockets por meio das fontes de dados de ponte. Exigem o sgcWebSockets instalado e compilam do Delphi 2010 ao Delphi 13, Win32 e Win64. Os clientes, servidores sgcWebSockets e as fontes de dados são colocados no formulário em tempo de design.

Demonstração	O que mostra	Pasta
MQTTSensors	Três componentes **TsgcChartDataSource_MQTT** compartilham um `TsgcWSPClient_MQTT` e exibem os tópicos `sgccharts/sensors/temperature`, `humidity` e `pressure`, assinados uma única vez em `sgccharts/sensors/+`, contra test.mosquitto.org via WebSocket ou um broker local via TCP simples; um botão Simular publica valores aleatórios a cada 100 ms para que a demonstração funcione com qualquer broker, e a pressão ganha seu próprio eixo direito.	Delphi\MQTTSensors\
BinanceLive	Dois componentes **TsgcChartDataSource_Binance** em um único `TsgcWSAPI_Binance`: um stream de kline alimenta uma série Candlestick que atualiza o candle em formação a cada mensagem, e um stream de negociações alimenta uma série Line; os últimos 60 candles são carregados por meio da API REST da Binance antes de assinar, e o símbolo (BTCUSDT ou ETHUSDT) e o intervalo (1m ou 5m) podem ser trocados enquanto conectado.	Delphi\BinanceLive\
WebSocketFeed	Um `TsgcWebSocketServer` dentro da demonstração transmite `{"t":<unix ms>,"v":<value>}` a cada 20 ms, e um **TsgcChartDataSource_WebSocket** em um `TsgcWebSocketClient` o plota, lendo **XPath** `t` como milissegundos Unix e **YPath** `v`; servidor e cliente executam no mesmo programa loopback em 127.0.0.1.	Delphi\WebSocketFeed\
SSEFeed	Um stream de eventos servido dentro da demonstração, lido com `TsgcHTTP1Client.GetSSE` em uma thread de trabalho e plotado por um **TsgcChartDataSource_SSE**, que divide o stream bruto em eventos e lê seus dados JSON enquanto o próprio **OnSSEMessage** do formulário continua disparando e cancela o stream ao Parar.	Delphi\SSEFeed\

Demonstrações FMX

Esta demonstração coloca um TsgcFMXChart em um formulário FireMonkey. Compila com Delphi 12 e 13, para Win32, Win64, macOS (OSX64, OSXARM64), iOSDevice64 e Android64.

Demonstração	O que mostra	Pasta

FMX-Mobile	Um controle de abas com uma **TsgcChartSeries_FastLine** ao vivo (um buffer circular de 200.000 pontos alimentado por uma thread em segundo plano a 20.000 pontos por segundo) e uma **TsgcChartSeries_Candlestick** com 500 candles, dimensionada para toque; zoom por pinça e pan no eixo de tempo, a mira e a dica de ferramenta, um botão Reset, uma **TimeWindow** de 10 s / 30 s, uma alternância de tema claro/escuro, e quadros por segundo a partir de **TsgcFMXChart.FrameCount**.	Delphi\FMXMobile\

Demonstrações de Renderização Sem Interface Visual

Estas demonstrações renderizam gráficos sem controle visual algum, por meio de TsgcChartRenderer, que não tem dependência de VCL ou FMX e, portanto, não precisa de nenhum deles instalado. Executam em programas de console, serviços do Windows, módulos ISAPI e daemons Linux64.

Demonstração	O que mostra	Pasta
Headless	Um programa de console que monta uma série Line de 10.000 pontos e uma série Candlestick de 500 candles em memória e renderiza cada uma para um arquivo PNG e SVG (`line.png` , `line.svg` , `candles.png` , `candles.svg`), imprimindo o tamanho e o CRC32 de cada arquivo. Os dados são gerados por aritmética de inteiros e o rasterizador é idêntico bit a bit entre plataformas, de modo que os arquivos PNG gravados no Win64 e no Linux64 compartilham o mesmo CRC32. Executa em Win32, Win64 e Linux64.	Delphi\Headless\
HeadlessService	Um serviço do Windows, instalado com `HeadlessService.exe /install` e removido com `/uninstall` , que adiciona um valor a um gráfico de linha e o renderiza para `output\chart.png` a cada 10 segundos; como um serviço não tem área de trabalho, o gráfico é desenhado por **TsgcChartRenderer** em vez de um controle visual, e os erros de renderização vão para o log de eventos do Windows. Executa em Win32 e Win64.	Delphi\HeadlessService\

Veja Também

- Início Rápido
- Fontes de Dados
- Fontes de Dados de Ponte
- TsgcChartRenderer

TsgcChart

Unidade: sgcCharts_Chart
TsgcChart é o modelo do gráfico: uma descendente de `TComponent` que contém os eixos, as séries, o título, subtítulo, legenda e margens e o tema e paleta, mas não desenha nada por conta própria e não depende de nenhum controle visual. Uma view (um controle VCL ou FireMonkey, ou um renderizador headless) chama **AddListener** para saber quando o modelo foi alterado e **Render** para pintá-lo em um TsgcUICanvas. Como o modelo e o desenho são separados, o mesmo TsgcChart pode ser exibido por mais de uma view ao mesmo tempo, ou renderizado diretamente para um arquivo PNG ou SVG sem nenhuma view.

Propriedades

Propriedade	Tipo	Descrição
Theme	TsgcUITheme	`utLight`, `utDark` ou `utCustom`. Padrão `utLight`. Veja Temas e Paleta.
Axes	TsgcChartAxes	A coleção de eixos. Veja Eixos.
Title, Subtitle	TsgcChartTitle	Veja Título, Legenda e Margens.
Legend	TsgcChartLegend	Veja Título, Legenda e Margens.
Margins	TsgcChartMargins	Veja Título, Legenda e Margens.
BackgroundColor, PlotBackgroundColor	TsgcUIColor	Armazenado apenas quando **Theme** é `utCustom`. Definir qualquer um deles diretamente muda **Theme** para `utCustom`, a menos que o componente esteja sendo carregado de um stream.
Palette	TsgcChartPalette	As 8 cores automáticas das séries. Veja Temas e Paleta.
MaxFPS	Integer	Limite de repintura que as views devem respeitar. Padrão 60.
TimeWindow	Double	Segundos, padrão 0 (desativado). Enquanto diferente de zero, um eixo inferior numérico ou de data e hora automático exibe `[newest X - TimeWindow, newest X]` em vez do intervalo completo de dados.
PlotRect	TsgcUIRect (somente leitura)	O retângulo de plotagem calculado pelo último **Render**.
Scale	Double	Fator HiDPI. A view o define antes de chamar **Render**; alterá-lo não gera notificação.
SeriesCount	Integer (somente leitura)	Número de séries cuja propriedade **Chart** aponta para este gráfico.
Series[Index: Integer]	TsgcChartSeries (so-	A série em Index, baseada em 0.

	mente leitura, default)	

Métodos

Método	Retorno	Descrição
BeginUpdate / EndUpdate		Agrupa várias alterações para que os listeners sejam notificados uma única vez, em **EndUpdate**, em vez de após cada alteração.
Invalidate		Notifica os listeners imediatamente, ou adia a notificação para o **EndUpdate** correspondente quando uma atualização está em aberto.
AddListener(aEvent: TNotifyEvent)		Registra um evento que uma view chama de volta quando o gráfico precisa ser repintado.
RemoveListener(aEvent: TNotifyEvent)		Remove o registro de um listener adicionado com **AddListener**.
Render(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect)		Pinta o plano de fundo, o título, o subtítulo, a legenda, os eixos e todas as séries visíveis dentro de **aRect** (pixels de dispositivo) de **aCanvas**. Veja Canvas.
FindAxis(const aName: string)	TsgcChartAxis	O eixo com esse Name (sem diferenciar maiúsculas de minúsculas), nil quando nenhum.
AxisRect(aAxis: TsgcChartAxis)	TsgcUIRect	A faixa que o último **Render** reservou para aAxis fora da área de plotagem; um retângulo vazio quando desconhecido.
AreaCount, AreaRect(aIndex), AreaMode(aIndex), AreaAt(X, Y)	Integer, TsgcUIRect, TsgcChartPlotMode, Integer	As áreas de plotagem da última **Render**, seus retângulos e modos, e a área sob um pixel. Veja Áreas de Plotagem.
SeriesAreaRect(aSeries), IsSeriesDrawn(aSeries)	TsgcUIRect, Boolean	Onde a última **Render** desenhou uma série, e se chegou a desenhá-la.

Modelo e views

TsgcChart não carrega nenhuma superfície de desenho própria: é um componente simples, não um controle, de modo que pode ser construído, populado e renderizado a partir de um serviço ou de uma aplicação de console exatamente como a partir de um formulário. As séries são elas próprias componentes com uma propriedade **Chart** (veja TsgcChartSeries); atribuir essa propriedade registra a série no gráfico, e limpá-la (ou destruir a série) a remove. Uma view só precisa de três coisas do modelo: **AddListener** para ser avisada de quando repintar, **Render** para fazer a pintura, e **Scale** definida com seu fator HiDPI de antemão.

Atualizações e notificações

Cada alteração no gráfico, em suas séries ou em seus eixos chama **Invalidate**, que notifica imediatamente os listeners registrados. Envolver várias alterações entre **BeginUpdate** e **EndUpdate** adia essa notificação até o **EndUpdate** correspondente, de modo que uma view repinta uma única vez em vez de uma vez por alteração.

Exemplo: Construindo e renderizando um gráfico em 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;
```

Consulte também

- TsgcChartSeries
- Eixos

- Áreas de Plotagem
- Canvas

TsgcChartSeries

Unidade: sgcCharts_Chart
TsgcChartSeries é a base abstrata de toda série: uma descendente de `TComponent` que um TsgcChart exibe por meio de sua propriedade **Chart**. Ela define a cor, a visibilidade, a legenda e as propriedades de vinculação a eixo que toda série compartilha, e dois membros abstratos, **GetDataRange** e **Draw**, que um tipo concreto implementa. Os tipos concretos de série (linha, área, linha em degraus, barra, dispersão, candlestick, OHLC e os demais) são componentes por direito próprio, declarados em `sgcCharts_Series` e documentados separadamente.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O gráfico que exibe esta série. Atribuí-la registra a série no gráfico; limpá-la, ou defini-la para um gráfico diferente, primeiro a remove do registro do gráfico anterior.
Title	string	Texto exibido na legenda. Veja **LegendText**.
Color	TsgcUIColor	0 (o padrão) significa uma cor automática da paleta. Veja **EffectiveColor**.
Visible	Boolean	Padrão True.
ShowInLegend	Boolean	Padrão True.
AxisX, AxisY	string	O **Name** do eixo contra o qual esta série é plotada. Veja Eixos.
Decimation	TsgcChartDecimation	`dmNone` , `dmMinMax` ou `dmLTTB` . Padrão `dmMinMax` . Veja Decimação.
Index	Integer (somente leitura)	Posição da série na lista de séries do gráfico, -1 quando não vinculada a um gráfico.

Métodos

Método	Retorno	Descrição
Changed		Pede a **Chart** para repintar (chama seu **Invalidate**); não faz nada quando **Chart** é nil.
EffectiveColor	TsgcUIColor	**Color**, ou a cor da paleta do gráfico em **Index** quando **Color** é 0.
XAxis	TsgcChartAxis	O eixo indicado por **AxisX**, resolvido em **Chart**; o primeiro eixo inferior quando **AxisX** está vazio, nil quando não encontrado.

YAxis	TsgcChartAxis	O eixo indicado por **AxisY**, resolvido em **Chart**; o primeiro eixo esquerdo quando **AxisY** está vazio, nil quando não encontrado.
GetDataRange(var aXMin, aXMax, aYMin, aYMax: Double)	Boolean (abstrato)	A extensão de dados da série; False quando não há dados. Implementado pelo tipo concreto de série.
Draw(aCanvas: TsgcUICanvas; aXAxis, aYAxis: TsgcChartAxis; const aPlot: TsgcUIRect; aScale: Double)		Abstrato. Pinta a série dentro de **aPlot**, já recortada (clipped) para ele; os valores são mapeados com **aXAxis.ValueToPixel** e **aYAxis.ValueToPixel**.
DrawLegendSymbol(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Pinta o símbolo da legenda. O padrão é um quadrado preenchido com **EffectiveColor**; os tipos concretos o sobrescrevem.
LegendText	string	**Title** quando definido, senão o **Name** do componente.
PlotMode	TsgcChartPlotMode (virtual)	Modo que esta série dá à sua área de plotagem: `cpmXY` (padrão), `cpmPolar` ou `cpmCell`. Veja Áreas de Plotagem.
DrawArea(aCanvas; const aRect; const aSeries: TsgcChartSeriesArray; aCount: Integer; aScale: Double)	(virtual)	Pinta uma área polar (chamado no primeiro radar da área, com todos os radares desenhados em **aSeries**) ou uma célula (**aSeries** contém apenas esta série). O canvas é recortado para a área. O padrão não faz nada.
HitTestArea(const aRect; const aSeries; aCount: Integer; X, Y: Double; var aHit: TsgcChartHitInfo)	Boolean (virtual)	Teste de acerto de uma área polar ou de uma célula, com os argumentos de **DrawArea**. O padrão retorna False.
HitTestPoint(const aPlot: TsgcUIRect; X, Y: Double; var aHit: TsgcChartHitInfo)	Boolean (virtual)	Teste de acerto de uma série XY que a interação não conhece, como o heatmap. O padrão retorna False.
TooltipText(const aHit: TsgcChartHitInfo)	string (virtual)	Dica de ferramenta de um acerto desta série; uma string vazia (o padrão) usa o formato da interação. Várias linhas são desenhadas como linhas simples.
AreaClick(const aHit: TsgcChartHitInfo)	(virtual)	Um clique em um acerto desta série, depois de **OnSeriesClick**. Explodir (pizza) e detalhar (treemap) rodam aqui. O padrão não faz nada.
LegendItemCount	Integer (virtual)	Entradas de legenda da série, 1 por padrão; uma pizza retorna uma por fatia.
MeasureLegendItem(aCanvas; aItem: Integer; const aFont; aScale: Double; out aWidth, aHeight: Double)	(virtual)	Tamanho da entrada de legenda **aItem**. O padrão é o símbolo, um espaço e **LegendText** em uma linha.
DrawLegendItem(aCanvas; aItem: Integer; const aRect; const aFont; aScale: Double)	(virtual)	Pinta a entrada de legenda **aItem** dentro de **aRect**. O padrão chama **DrawLegendSymbol** e desenha **LegendText**.
GetDataCategories(aAxis: TsgcChartAxis; aList: TStrings)	Boolean (virtual)	Textos de categoria em tempo de execução para um eixo de categoria sem **Categories**; True quando **aList** foi preenchida. O heatmap o usa para os rótulos de suas linhas e células. O padrão retorna False.

Vinculação ao gráfico e aos eixos

Uma série não pertence ao seu gráfico: definir **Chart** apenas a registra, de modo que a aplicação continua responsável por liberar a instância da série (ou dá a ela um owner da forma usual do Delphi). **AxisX** e **AxisY** vinculam a série a um dos eixos em **Chart.Axes** por **Name** (sem diferenciar maiúsculas de minúsculas); um nome vazio recorre ao primeiro eixo inferior para X e ao primeiro eixo esquerdo para Y, de modo que um gráfico com um eixo de cada não exige que toda série os nomeie.

Cor e a paleta

Deixar **Color** em 0 faz a série usar a cor da **Palette** do gráfico correspondente ao seu **Index**, percorrendo em ciclo as 8 cores da paleta. Definir **Color** com um valor diferente de zero sobrescreve a paleta para essa série específica. Veja Temas e Paleta.

Decimation

Decimation controla como uma série concreta reduz um buffer grande para aproximadamente um conjunto de pontos por coluna de pixel antes de desenhar. Veja Decimação para saber o que cada modo preserva.

Hooks da área de plotagem

Todo hook assume por padrão o comportamento XY, de modo que uma série que sobrescreve apenas **GetDataRange** e **Draw** continua sendo uma série XY. As séries pie, radar, funnel e treemap sobrescrevem **PlotMode**, **DrawArea** e **HitTestArea**; o heatmap continua sendo uma série XY e sobrescreve **HitTestPoint** e **GetDataCategories**. Veja Áreas de Plotagem para saber como um modo é escolhido.

Exemplo: Uma série de linha vinculada a eixos nomeados

```
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;
```

Consulte também

- TsgcChart
- Eixos
- Decimação
- Áreas de Plotagem

TsgcChartAxis e TsgcChartAxes

Unidade: sgcCharts_Axis
TsgcChartAxes é a `TOwnedCollection` exposta como TsgcChart.**Axes**; cada item é um **TsgcChartAxis** (um `TCollectionItem`). Um gráfico pode conter mais de um eixo do mesmo lado, o que permite que um gráfico exiba duas escalas de valores ao mesmo tempo: as séries se vinculam a um eixo cada por nome (veja TsgcChartSeries).

Propriedades de TsgcChartAxis

Propriedade	Tipo	Descrição
Name	string	Identificador ao qual as séries se vinculam por meio de **AxisX** / **AxisY**, e que **TsgcChart.FindAxis** procura.
AxisType	TsgcChartAxisType	`catNumeric` (padrão), `catLogarithmic`, `catDateTime` ou `catCategory`.
Position	TsgcChartAxisPosition	`capLeft` (padrão), `capRight`, `capTop` ou `capBottom`. Os eixos superior e inferior são horizontais (X); os eixos esquerdo e direito são verticais (Y). Veja **IsHorizontal**.
Title, TitleStyle	string, TsgcUITextStyle	Título do eixo e seu estilo de texto.
LabelStyle	TsgcUITextStyle	Estilo de texto dos rótulos das marcas.
Visible	Boolean	Padrão True.
Automatic	Boolean	Padrão True. Quando True, o intervalo vem dos dados das séries vinculadas a este eixo. Quando False, **Minimum** / **Maximum** são usados conforme informados.
Minimum, Maximum	Double	Intervalo manual, usado quando **Automatic** é False.
Inverted	Boolean	Padrão False.
LabelFormat	string	Vazio para formatação automática. Para `catNumeric` / `catLogarithmic`, uma máscara `FormatFloat` independente do locale (`'0.00'`, `'#,##0.0'`). Para `catDateTime`, as letras de formato de data e hora abaixo.
GridVisible, GridColor, LineColor	Boolean, TsgcUIColor	Visibilidade das linhas de grade e as cores da grade e da linha do eixo. **GridVisible** padrão True.
TickLength	Double	Comprimento da marca de escala em pixels de dispositivo na escala 1.
MinTickSpacing	Double	Espaçamento mínimo em pixels de dispositivo, na escala 1, entre duas marcas. Padrão 50.
Categories	TStrings	Rótulos de marca para um eixo `catCategory`, um por posição.

Métodos de TsgcChartAxis

Método	Retorno	Descrição
IsHorizontal	Boolean	True quando **Position** é `capTop` ou `capBottom`.
SetDataRange(aMin, aMax: Double; aExact: Boolean = False)		Informa a extensão dos dados a partir da qual um eixo automático deve se expandir para arredondar valores; **aExact** exibe **[aMin, aMax]** exatamente como informado, sem expansão (usado para **TimeWindow**).
ClearDataRange		Limpa a extensão definida por **SetDataRange**.
Textos de categoria em tempo de execução de um eixo de categoria sem **Categories**; o gráfico os define durante a renderização a partir de **GetDataCategories** das séries, como **SetDataRange**. Eles nunca são gravados no stream.		Run-time category texts of a category axis without **Categories**; the chart sets them during the render from **GetDataCategories** of the series, like **SetDataRange**. They are never streamed.
ClearDataCategories		Limpa os textos definidos por **SetDataCategories**.
ZoomRange(aMin, aMax: Double)		Restringe o intervalo visível a **[aMin, aMax]**.
ResetZoom		Limpa um intervalo definido por **ZoomRange**.
Zoomed	Boolean (somente leitura)	True enquanto um intervalo de zoom está ativo.
VisibleMin, VisibleMax	Double (somente leitura)	O intervalo atualmente em vigor: o intervalo de zoom, senão o intervalo manual ou o intervalo de dados.
Calculate(aCanvas: TsgcUICanvas; aPixelLength: Double; aScale: Double = 1)		Calcula as marcas para um trecho de **aPixelLength** pixels ao longo deste eixo. **aCanvas** mede os rótulos para o afinamento e pode ser nil.
TickCount, TickValue[Index], TickLabel[Index]	Integer, Double, string	As marcas calculadas pelo último **Calculate**. **TickLabel** fica vazio para um rótulo removido pelo afinamento.
Step, DateTimeUnit	Double, TsgcChartDateTimeUnit	O espaçamento entre marcas e, para um eixo `catDateTime`, a unidade para a qual esse espaçamento foi arredondado.
ValueToPixel(aValue, aStart, aLength: Double)	Double	Converte um valor de dado em uma posição de pixel do dispositivo.
PixelToValue(aPixel, aStart, aLength: Double)	Double	O inverso de **ValueToPixel**.

MeasureSize(aCanvas: TsgcUICanvas; aScale: Double = 1)	Double	Espessura que este eixo precisa fora da área de plotagem (marcas, rótulos, título); 0 quando não está **Visible**.
Draw, DrawGrid		Desenha o eixo, ou suas linhas de grade, para o último **Calculate**.

Membros de TsgcChartAxes

Membro	Descrição
Add: TsgcChartAxis	Adiciona um novo eixo e o retorna.
Find(const aName: string): TsgcChartAxis	Primeiro eixo com esse Name (sem diferenciar maiúsculas de minúsculas), nil quando nenhum.
Items[Index: Integer]: TsgcChartAxis (default)	Eixo em Index.

Intervalo automático e zoom

Enquanto **Automatic** é True, o gráfico alimenta o eixo com a extensão combinada dos dados de cada série vinculada a ele a cada renderização, por meio de **SetDataRange**; um eixo automático expande esse intervalo para arredondar os valores das marcas, a menos que o gráfico chame **SetDataRange** com **aExact** True, o que ele faz para **TsgcChart.TimeWindow**. Definir **Automatic** como False e informar **Minimum** / **Maximum** fixa o intervalo em vez disso. **ZoomRange** e **ResetZoom** se aplicam sobre qualquer um dos dois, para zoom interativo; **VisibleMin** / **VisibleMax** informam o intervalo que está de fato em vigor.

Formato de rótulo

Para um eixo `catDateTime`, **LabelFormat** usa letras de data e hora independentes de locale, independentemente do locale em tempo de execução: `yyyy`, `yy`, `mmmm`, `mmm`, `mm`, `m`, `dd`, `d`, `hh`, `h`, `nn`, `n`, `ss`, `s`, `zzz`. O texto entre aspas é copiado como está. `mm` é sempre o mês; use `nn` para minutos. Deixado vazio, o eixo escolhe automaticamente um formato a partir da **DateTimeUnit** do espaçamento das marcas.

Categories

Um eixo `catCategory` coloca uma marca por entrada de **Categories**, na ordem da lista, em vez de calcular marcas numéricas; as séries vinculadas a ele são plotadas em posições inteiras correspondentes a essas entradas. Quando **Categories** está vazio, uma série pode fornecer os textos em tempo de execução: um heatmap rotula suas linhas e células dessa forma.

Múltiplos eixos de valor

Um gráfico não se limita a um eixo X e um eixo Y: adicionar um segundo eixo com o mesmo **Position** de um já existente, e apontar algumas séries para ele por meio de **AxisX** / **AxisY**, dá a essas séries sua própria escala, enquanto as demais continuam usando o primeiro eixo. Uma série que deixa **AxisX** ou **AxisY** vazio se vincula, respectivamente, ao primeiro eixo inferior e ao primeiro eixo esquerdo.

Exemplo: Um eixo inferior de data e hora e um eixo esquerdo 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;
```

Consulte também

- TsgcChart
- TsgcChartSeries

Título, Legenda e Margens

Unidade: sgcCharts_Chart
Três pequenas classes `TPersistent` contêm a decoração ao redor do gráfico: **TsgcChartTitle** (usada tanto para TsgcChart.**Title** quanto para **Subtitle**), **TsgcChartLegend** (**TsgcChart.Legend**) e **TsgcChartMargins** (**TsgcChart.Margins**). Todas as três disparam uma notificação interna de alteração que invalida o gráfico, de modo que editá-las em tempo de execução repinta automaticamente.

Propriedades de TsgcChartTitle

Propriedade	Tipo	Descrição
Text	string	O texto do título. Um **Text** vazio ainda assim não reserva espaço; o título é omitido quando está vazio ou não está **Visible**.
TextStyle	TsgcUITextStyle	Tamanho, negrito e cor da fonte. Veja abaixo.
Visible	Boolean	Padrão True.
Alignment	TAlignment	Padrão `taCenter`.

Propriedades de TsgcChartLegend

Propriedade	Tipo	Descrição
Visible	Boolean	Padrão True.
Position	TsgcChartLegendPosition	`clpTop` (padrão), `clpBottom`, `clpLeft` ou `clpRight`.
TextStyle	TsgcUITextStyle	Fonte das entradas da legenda.
BackgroundColor	TsgcUIColor	Padrão 0 (alfa 0, transparente).
Padding	Integer	Padrão 4, pixels de dispositivo na escala 1.

Uma série aparece na legenda quando sua própria **ShowInLegend** é True (veja TsgcChartSeries); o texto da legenda é o **LegendText** da série.

Propriedades de TsgcChartMargins

Propriedade	Tipo	Descrição
Left, Top, Right, Bottom	Integer	Margem externa em pixels de dispositivo, na escala 1, ao redor de todo o gráfico, fora do título, da legenda e dos eixos. Padrão 10 em todos os lados.

Propriedades de TsgcUITextStyle

Unidade: sgcCharts_Types. Usado por **TsgcChartTitle.TextStyle**, **TsgcChartLegend.TextStyle** e **TsgcChartAxis.TitleStyle** / **LabelStyle**.

Propriedade	Tipo	Descrição
Size	Double	Tamanho da fonte em pontos. Padrão 11.
Bold	Boolean	Padrão False.
Color	TsgcUIColor	

ToFontStyle(aScale: Double = 1) retorna o **TsgcUIFontStyle** correspondente para uma chamada de canvas, com **Size** multiplicado por **aScale**.

Itens de legenda e barras de cores

Uma série pode acrescentar várias entradas à legenda, e as linhas da legenda assumem a altura de sua entrada mais alta. Uma pizza acrescenta uma entrada por fatia desenhada, Others por último; um funil uma por etapa desenhada; um treemap no modo `ctcGroup` uma por grupo de nível superior. Um heatmap, e um treemap no modo `ctcValue`, acrescentam uma barra de cores com o mínimo e o máximo de sua escala de cores, horizontal ou vertical conforme a posição da legenda. **ShowInLegend** False oculta todas as entradas de uma série, e uma série ignorada por sua área de plotagem não mostra nenhuma.

Exemplo: Título, subtítulo, posição da legenda e margens

```
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;
```

Consulte também

- TsgcChart
- TsgcChartSeries

Temas e Paleta

Unidade: sgcCharts_Chart (TsgcChartPalette), sgcCharts_Types (TsgcUITheme)
TsgcChart.**Theme** define de uma só vez o plano de fundo, o texto, a grade e as cores de linha de eixo padrão embutidos, além da paleta de séries padrão; **Palette** é a lista de 8 cores que as séries usam automaticamente por índice.

TsgcUITheme

Valor	Descrição
utLight	Padrão. Plano de fundo claro, texto escuro, grade clara.
utDark	Plano de fundo escuro, texto claro, grade escura.
utCustom	Deixa **BackgroundColor**, **PlotBackgroundColor** e **Palette** exatamente como estão; o gráfico não aplica nenhuma cor embutida.

`utLight` e `utDark` possuem, cada um, sua própria **Palette** padrão (8 cores, usadas na ordem de índice das séries). Definir **BackgroundColor**, **PlotBackgroundColor** ou qualquer cor de **Palette** diretamente muda **Theme** automaticamente para `utCustom`, a menos que o gráfico esteja sendo carregado de um stream; a partir desse ponto, alterar **Theme** de volta para `utLight` ou `utDark` reaplica as cores desse tema e sobrescreve as cores personalizadas.

Propriedades de TsgcChartPalette

Propriedade	Tipo	Descrição
Colors[Index: Integer] (default)	TsgcUIColor	Index é considerado módulo 8, portanto nunca gera erro para um índice de série fora do intervalo.
Color1 .. Color8	TsgcUIColor	As 8 cores como propriedades publicadas, índice 0 a 7.

Métodos de TsgcChartPalette

Método	Descrição
LoadColors(const aColors: array of TsgcUIColor)	Copia até 8 cores de **aColors** para **Color1**.. **Color8**, em ordem.

Uma série com **Color** = 0 (o padrão) usa **Palette.Colors[Series.Index]** por meio de **EffectiveColor**; veja TsgcChartSeries.

Exemplo: Tema escuro e depois uma 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;
```

Consulte também

- TsgcChart
- TsgcChartSeries

Áreas de Plotagem

Unidade: sgcCharts_Chart
Uma área de plotagem é o gráfico em si, ou um painel visível quando o gráfico tem painéis. A cada renderização, o gráfico dá a cada área um modo, calculado a partir das séries visíveis vinculadas a ela:

Modo	Séries	O que a área desenha
`cpmXY`	Séries de linha, área, barra, dispersão, candlestick, heatmap e as demais séries XY	Eixos, grade e todas as séries XY por meio de **Draw**.
`cpmPolar`	TsgcChartSeries_Radar	Uma grade polar; o primeiro radar desenha todos os radares da área sobre ela.
`cpmCell`	TsgcChartSeries_Pie, TsgcChartSeries_Funnel, TsgcChartSeries_Treemap	Uma grade de células, uma por série, cada uma com uma faixa de título; sem eixos.

A regra de mistura

XY prevalece sempre que uma série XY está visível na área. Caso contrário, a primeira série visível decide entre polar e célula. As séries que não correspondem ao modo de sua área são ignoradas: não são desenhadas, não passam por teste de acerto e não aparecem na legenda. Para mostrar uma pizza ao lado de um gráfico de linhas, dê ao gráfico dois painéis e vincule cada série ao eixo de seu painel, ou use dois gráficos.

Membros de área de TsgcChart

Método	Retorno	Descrição
AreaCount	Integer (somente leitura)	Áreas de plotagem da última **Render**: 1 para o gráfico, ou uma por painel visível na ordem da coleção.
AreaRect(aIndex: Integer)	TsgcUIRect	Retângulo de uma área, vazio quando **aIndex** está fora do intervalo.
AreaMode(aIndex: Integer)	TsgcChartPlotMode	Modo de uma área, `cpmXY` quando **aIndex** está fora do intervalo.
AreaAt(X, Y: Double)	Integer	Índice da área sob um pixel do gráfico, -1 quando nenhuma.
SeriesAreaRect(aSeries: TsgcChartSeries)	TsgcUIRect	A célula de uma série de célula desenhada, a área de uma série polar desenhada, o retângulo do gráfico do painel de uma série XY, vazio para qualquer outra série.
IsSeriesDrawn(aSeries: TsgcChartSeries)	Boolean	True quando a última **Render** desenhou a série: visível, do modo de sua área e, para uma série XY, com ambos os seus eixos.

As células são dispostas como uma grade que preenche a área. Com várias células, cada uma começa abaixo de uma faixa de título que contém o **Title** da série. A interação encaminha o hover, o clique e as dicas de ferramenta para as séries de todos os modos, mas o zoom, o deslocamento e a mira funcionam apenas em áreas XY. Um autor de série implementa os hooks listados em TsgcChartSeries.

Exemplo: Duas séries de célula e depois uma série 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.
```

Consulte também

- TsgcChart
- TsgcChartSeries
- Painéis
- TsgcChartInteraction

Escala de Cores

Unidade: sgcCharts_ColorScale
TsgcChartColorScale mapeia um valor para uma cor por meio de uma tabela de consulta (LUT) de 256 entradas construída a partir de duas ou três paradas de cor. É a propriedade **ColorScale** de TsgcChartSeries_Heatmap e de TsgcChartSeries_Treemap no modo `ctcValue`, e a legenda a mostra como uma barra de cores com o mínimo e o máximo.

Propriedades

Propriedade	Tipo	Descrição
MinColor	TsgcUIColor	Cor do mínimo; 0 (padrão) usa o fundo do gráfico do tema.
MidColor	TsgcUIColor	Cor do meio do intervalo; 0 (padrão) significa duas paradas.
MaxColor	TsgcUIColor	Cor do máximo; 0 (padrão) usa a cor da série ou, na falta dela, a primeira cor da paleta.
Steps	Integer	0 (padrão) é uma escala contínua; de 2 a 32 é uma paleta em degraus com essa quantidade de faixas.
Automatic	Boolean	Padrão True: o intervalo é o mínimo e o máximo das células ou dos blocos visíveis. False usa **Minimum** e **Maximum**.
Minimum, Maximum	Double	O intervalo fixo. Padrão 0 e 100.
NullColor	TsgcUIColor	Cor das células NaN; 0 (padrão) as deixa sem pintar.
RangeMin, RangeMax	Double (somente leitura)	O intervalo definido pela última renderização.
LUT[aIndex]	TsgcUIColor (somente leitura)	Uma entrada da tabela de consulta, de 0 a 255.
OnChange	TNotifyEvent	Dispara quando uma propriedade muda; a série proprietária redesenha.

Métodos

Método	Retorno	Descrição
Prepare(aDefaultMinColor, aDefaultMaxColor: TsgcUIColor; const aDataMin, aDataMax: Double)		Chamado pela série antes de desenhar: resolve as cores zero, reconstrói a LUT quando uma parada ou **Steps** mudou e define o intervalo. Um intervalo de largura zero mapeia todo valor para a entrada 128.
IndexOf(const aValue: Double)	Integer	Entrada da LUT de um valor, de 0 a 255; -1 para NaN.

ColorOf(const aValue: Double)	TsgcUIColor	Cor da LUT de um valor; **NullColor** para NaN.
DrawBar, MeasureColorBar, DrawColorBar		Desenham a barra e a barra de cores da legenda com seus rótulos; usados pelos itens de legenda da série.

Exemplo: Uma escala de semáforo em degraus em um intervalo fixo

```
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.
```

Consulte também

- TsgcChartSeries_Heatmap
- TsgcChartSeries_Treemap
- Título, Legenda e Margens
- Temas e Paleta

TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine

Unidade: sgcCharts_Series
Essas três séries descendem de **TsgcChartSeries_XY**, a base comum de toda série X/Y (linha, linha rápida, linha em degraus, área, barra e dispersão, veja TsgcChartSeries para as propriedades que toda série compartilha). TsgcChartSeries_XY possui o buffer de pontos e os métodos que o alimentam; **TsgcChartSeries_Line** desenha uma polilinha com um padrão de traço opcional e marcadores de ponto; **TsgcChartSeries_FastLine** é uma variante mais fina destinada a grandes volumes de dados ao vivo; **TsgcChartSeries_StepLine** descende de TsgcChartSeries_Line e desenha uma linha em degraus em vez de segmentos retos.

TsgcChartSeries_XY

TsgcChartSeries_XY possui um **Buffer** de pontos X/Y e implementa os métodos de dados que toda série de linha, área, linha em degraus, barra e dispersão compartilha. Não é abstrata, mas normalmente é usada por meio de uma de suas descendentes em vez de diretamente.

Propriedade	Tipo	Descrição
Buffer	TsgcChartBuffer (somente leitura)	O buffer circular subjacente. Veja TsgcChartBuffer e TsgcChartOHLCBuffer.
Capacity	Integer	Número máximo de pontos que o buffer mantém. Padrão 100000.
BufferMode	TsgcChartBufferMode	`cbmOverwrite` (padrão) descarta o ponto mais antigo assim que o buffer fica cheio; `cbmGrow` dobra a capacidade em vez disso.

Métodos

Método	Retorna	Descrição
AddXY(const aX, aY: Double)		Adiciona um ponto e chama **Changed**.
AddY(const aY: Double)		Adiciona um ponto em X igual ao X do ponto mais novo mais 1 (0 para um buffer vazio) e chama **Changed**. Útil quando X é um índice corrente em vez de uma coordenada real.
Clear		Esvazia o buffer e chama **Changed**.

TsgcChartSeries_Line

TsgcChartSeries_Line desenha os pontos do buffer como uma polilinha, com um padrão de traço opcional e marcadores de ponto opcionais.

Propriedade	Tipo	Descrição

LineWidth	Double	Pixels de dispositivo em escala 1. Padrão 2.
Dash	TsgcChart-DashStyle	`cdsSolid`, `cdsDash`, `cdsDot` ou `cdsDashDot`. Padrão `cdsSolid`.
Markers	TsgcChartSeri-esMarkers	Oculto por padrão (**Visible** é False). Veja TsgcChartSeriesMarkers abai-xo.

TsgcChartSeries_FastLine

TsgcChartSeries_FastLine é uma linha fina destinada a grandes volumes de dados ao vivo: não tem marcadores nem padrão de traço, e seus segmentos se unem com uma junção em bisel em vez da junção arredondada usada por TsgcChartSeries_Line, o que a mantém barata para redesenhar em uma taxa de atualização alta.

Propriedade	Tipo	Descrição
LineWidth	Double	Pixels de dispositivo em escala 1. Padrão 1.

TsgcChartSeries_StepLine

TsgcChartSeries_StepLine descende de TsgcChartSeries_Line, portanto também tem **LineWidth**, **Dash** e **Markers**, mas conecta pontos consecutivos com um segmento horizontal e um vertical em vez de uma diagonal reta.

Propriedade	Tipo	Descrição
StepMode	TsgcChartStepMode	`csmBefore`, `csmAfter` ou `csmCenter`. Padrão `csmAfter`.

Com **csmAfter**, um valor se mantém até que o próximo ponto seja alcançado, de modo que o degrau ocorre exatamente em cada ponto. Com **csmBefore**, o valor é alcançado no ponto anterior, de modo que o degrau ocorre logo antes dele. Com **csmCenter**, o degrau ocorre no meio do caminho entre os dois pontos.

TsgcChartSeriesMarkers

TsgcChartSeriesMarkers é o pequeno objeto persistente exposto pela propriedade **Markers** de TsgcChartSeries_Line e de TsgcChartSeries_Scatter. Ele desenha uma forma centralizada em cada ponto visível.

Proprie-dade	Tipo	Descrição
Visible	Boolean	Se os marcadores são desenhados. O padrão depende da série proprietária: False para TsgcChartSeries_Line e suas descendentes, True para TsgcChartSeries_Scatter.
Shape	TsgcChart-Mar-kerShape	`cmsSquare`, `cmsCircle`, `cmsTriangle` ou `cmsDiamond`. Padrão `cmsCircle`.
Size	Integer	Diâmetro em pixels de dispositivo em escala 1. Padrão 6.

Exemplo: séries de linha, linha rápida e linha em degraus em um único 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.
```

Veja Também

- TsgcChart
- TsgcChartSeries
- TsgcChartSeries_Area
- TsgcChartSeries_Scatter

TsgcChartSeries_Area

Unidade: sgcCharts_Series
TsgcChartSeries_Area descende de TsgcChartSeries_Line, portanto desenha a mesma polilinha (com **LineWidth**, **Dash** e **Markers**) e adicionalmente preenche a área entre essa linha e um valor de base.

Propriedades

Propriedade	Tipo	Descrição
Baseline	Double	O preenchimento vai da linha para baixo (ou para cima, quando a linha estiver abaixo dela) até este valor. Padrão 0.
FillOpacity	Double	0..1. Padrão 0.35.
Gradient	Boolean	False (padrão) preenche com uma cor sólida em **FillOpacity**. True preenche com um gradiente vertical que vai da cor de preenchimento em **FillOpacity** até totalmente transparente em **Baseline**.

Exemplo: uma série de área com uma linha de base elevada e um preenchimento em gradiente

```
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.
```

Veja Também

- TsgcChart
- TsgcChartSeries
- TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine

TsgcChartSeries_Bar

Unidade: sgcCharts_Series
TsgcChartSeries_Bar descende de **TsgcChartSeries_XY** (veja TsgcChartSeries_XY para **Buffer**, **AddXY**, **AddY**, **Clear**, **Capacity** e **BufferMode**) e desenha uma barra por ponto.

Propriedades

Propriedade	Tipo	Descrição
Decimation	TsgcChartDecimation	Padrão `dmNone` para uma série de barras, diferente do padrão `dmMinMax` da base TsgcChartSeries: as barras devem ser exibidas uma por coluna de pixel ou mais largas, sem serem decimadas. Veja Decimação.
Orientation	TsgcChartBarOrientation	`cboVertical` (padrão) ou `cboHorizontal`. Uma série de barras horizontal mapeia seu X (posição) para o eixo vertical e seu Y (valor) para o eixo horizontal, de modo que as barras crescem para os lados em vez de para cima.
BarWidthPercent	Integer	Parcela do slot X ocupada pela barra, ou pelo grupo de barras, 1..100. Padrão 70.
Grouping	TsgcChartBarGrouping	`cbgNone` (padrão), `cbgGrouped` ou `cbgStacked`. Veja abaixo.

Agrupamento e empilhamento

Com **cbgGrouped**, as séries de barras visíveis que compartilham os mesmos eixos, orientação e agrupamento são colocadas lado a lado dentro de cada slot X. Com **cbgStacked**, essas séries são empilhadas, na ordem das séries: valores positivos são empilhados no lado positivo e valores negativos no lado negativo, de modo que uma série com um valor negativo em um dado X não empurra a pilha positiva para cima. **cbgNone** desenha cada série de barras isoladamente, sobrepondo as demais no mesmo X.

Exemplo: duas séries 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.
```

Veja Também

- TsgcChart
- TsgcChartSeries
- Decimação

TsgcChartSeries_Scatter

Unidade: sgcCharts_Series
TsgcChartSeries_Scatter descende de **TsgcChartSeries_XY** (veja TsgcChartSeries_XY para **Buffer**, **AddXY**, **AddY**, **Clear**, **Capacity** e **BufferMode**) e desenha seus pontos apenas como marcadores, sem linha conectando-os.

Propriedades

Propriedade	Tipo	Descrição
Decimation	TsgcChartDecimation	Padrão `dmNone` para uma série de dispersão, diferente do padrão `dmMinMax` da base TsgcChartSeries: um gráfico de dispersão não tem linha para manter visualmente exata, portanto todo ponto é mantido. Veja Decimação.
Markers	TsgcChartSeriesMarkers	Visível por padrão (diferente do Markers da série de linha). Veja TsgcChartSeriesMarkers para **Shape** e **Size**; definir **Visible** como False oculta os pontos.

Exemplo: uma série de dispersão com 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.
```

Veja Também

- TsgcChart
- TsgcChartSeries
- TsgcChartSeries_Line, TsgcChartSeries_FastLine, TsgcChartSeries_StepLine
- Decimação

TsgcChartSeries_Candlestick, TsgcChartSeries_OHLC

Unidade: sgcCharts_Series
Essas duas séries descendem de **TsgcChartSeries_OHLCBase**, a base comum de toda série de abertura/máxima/mínima/fechamento. TsgcChartSeries_OHLCBase possui um buffer OHLC em vez do buffer X/Y usado pelas séries da família de linhas, e implementa os métodos que o alimentam; **TsgcChartSeries_Candlestick** desenha cada barra como um corpo de candle preenchido ou vazado com pavios, e **TsgcChartSeries_OHLC** desenha cada barra como uma linha vertical com marcas de abertura e fechamento.

TsgcChartSeries_OHLCBase

Propriedade	Tipo	Descrição
OHLCBuffer	TsgcChartOHLCBuffer (somente leitura)	O buffer circular subjacente de candles. Veja TsgcChartBuffer e TsgcChartOHLCBuffer.
Capacity	Integer	Número máximo de candles que o buffer mantém. Padrão 100000.
BufferMode	TsgcChartBufferMode	`cbmOverwrite` (padrão) descarta o candle mais antigo assim que o buffer fica cheio; `cbmGrow` dobra a capacidade em vez disso.
UpColor	TsgcUIColor	Cor usada quando o fechamento de um candle é maior ou igual à sua abertura. Padrão um verde, $FF1BAF7A.
DownColor	TsgcUIColor	Cor usada quando o fechamento de um candle é menor que sua abertura. Padrão um vermelho, $FFE34948.

Métodos

Método	Retorna	Descrição
AddOHLC(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)		Adiciona um candle e chama **Changed**.
UpdateLastOHLC(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)		Substitui o candle mais novo quando seu X é igual a **aX**, caso contrário adiciona um novo, e chama **Changed**. Destinado a um feed ao vivo que mantém a atualização do candle em formação até que seu período se encerre.
Clear		Esvazia o buffer e chama **Changed**.

TsgcChartSeries_Candlestick

TsgcChartSeries_Candlestick desenha cada candle como um corpo entre a abertura e o fechamento, colorido com **UpColor** ou **DownColor**, mais um pavio da máxima até a mínima.

Propriedade	Tipo	Descrição
WickColor	TsgcUI-Color	Padrão 0, o que significa que o pavio assume a cor do corpo do candle. Um valor diferente de zero o sobrepõe para todo candle.
BodyWidth-Percent	Integer	Parcela do slot do candle ocupada pelo corpo, 1..100. Padrão 70.
HollowUp	Boolean	Padrão False. True desenha os candles de alta (fechamento >= abertura) como um contorno em **UpColor** em vez de um corpo preenchido.

TsgcChartSeries_OHLC

TsgcChartSeries_OHLC desenha cada barra como uma linha vertical da máxima até a mínima, com uma pequena marca à esquerda na abertura e uma pequena marca à direita no fechamento, coloridas com **UpColor** ou **DownColor**.

Propriedade	Tipo	Descrição
TickWidthPer-cent	Inte-ger	Parcela do slot do candle ocupada pelas marcas de abertura e fechamento, 1..100. Padrão 70.
LineWidth	Dou-ble	Pixels de dispositivo em escala 1. Padrão 1.

Exemplo: uma série candlestick e uma série 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.
```

Veja Também

- TsgcChart
- TsgcChartSeries
- Panes, para dividir uma série candlestick em seu próprio painel abaixo do eixo de preço

TsgcChartSeries_Pie

Unidade: sgcCharts_Series_Pie
TsgcChartSeries_Pie desenha uma pizza ou, com **InnerRadiusPercent** acima de 0, uma rosca (donut). Ela descende de **TsgcChartSeries_Category** (unidade `sgcCharts_Series_Category`), a base de toda série que contém valores nomeados: um item por fatia, mantido na coleção **Slices**, que é também o editor em tempo de design e um armazenamento thread-safe para dados ao vivo. Uma pizza é uma série de célula: ela recebe sua própria célula da área de plotagem e não desenha eixos, veja Áreas de Plotagem.

Propriedades

Propriedade	Tipo	Descrição
Slices	TsgcChartCategoryItems	As fatias, uma **TsgcChartPieSlice** cada (**Text**, **Value**, **Color** e **Exploded**). As cores das fatias seguem o índice da coleção, de modo que uma fatia mantém sua cor com dados ao vivo.
InnerRadiusPercent	Double	0 (padrão) desenha uma pizza; até 90 desenha um anel de rosca cujo furo é essa porcentagem do raio.
StartAngle	Double	Ângulo da primeira borda de fatia em graus, 0 (padrão) é 12 horas.
Clockwise	Boolean	Padrão True.
Labels	TsgcChartSliceLabels	`cslNone`, `cslInside`, `cslOutside` (padrão) ou `cslAuto`. Veja Rótulos abaixo.
LabelFormat	string	Padrão `{text} ({percent})`. Os tokens são escritos como um nome entre chaves: `{text}`, `{value}` e `{percent}`.
ValueFormat	string	Máscara FormatFloat do token de valor; vazio (padrão) mostra até 2 casas decimais.
PercentDecimals	Integer	Casas decimais do token de porcentagem. Padrão 1.
LabelStyle	TsgcUITextStyle	Fonte dos rótulos. **Color** 0 usa a cor de texto do tema fora da pizza e texto branco ou escuro, escolhido pela cor da fatia, dentro dela.
LeaderLineColor	TsgcUIColor	Cor das linhas de chamada dos rótulos externos; 0 (padrão) usa a cor da fatia.
BorderColor	TsgcUIColor	Cor das bordas das fatias; 0 (padrão) usa a cor de fundo do gráfico.
BorderWidth	Double	Largura das bordas das fatias. Padrão 1.
ExplodeOffset	Integer	Distância em que uma fatia explodida é puxada para fora, em pixels de dispositivo na escala 1. Padrão 10.
ExplodeOnClick	Boolean	Padrão False. Quando True, um clique em uma fatia alterna seu flag **Exploded** (**OthersExploded** para a fatia Others).
OthersThreshold	Double	Uma porcentagem, 0 (padrão) desativa. As fatias abaixo dela são reunidas em uma única fatia Others, desenhada por último, quando pelo menos duas se qualificam.
OthersText	string	Texto da fatia Others. Padrão `Others`.

OthersColor	TsgcUIColor	Cor da fatia Others; 0 (padrão) usa a cor de texto esmaecida do tema.
OthersExploded	Boolean (public)	A fatia Others é puxada para fora da pizza.
Total	Double (somente leitura)	Soma dos valores desenhados na última renderização.

Métodos

Método	Retorno	Descrição
SliceColor(aIndex: Integer)	TsgcUIColor	A cor com que a fatia **aIndex** (um índice da coleção, ou **CS_CHART_OTHERS_INDEX**) é desenhada: sua **Color** ou, na falta dela, a cor da paleta do gráfico para **aIndex**. A fatia Others usa **OthersColor** ou, na falta dela, a cor de texto esmaecida do tema.

Métodos das séries de categoria

As séries pie, radar, funnel e treemap compartilham estes métodos de **TsgcChartSeries_Category**. Todos podem ser chamados de qualquer thread: o armazenamento tem seu próprio bloqueio, e uma gravação que não altera um valor não redesenha.

Método	Retorno	Descrição
SetValue(const aText: string; aValue: Double)	Integer	Define o valor do primeiro item com este texto (diferencia maiúsculas de minúsculas) ou acrescenta um novo item. Retorna o índice do item. O valor mais recente de um texto prevalece; os valores nunca são acumulados.
SetValueAt(aIndex: Integer; aValue: Double)		Define o valor do item em **aIndex**, acrescentando itens com textos vazios e valor 0 até **aIndex** quando a coleção é mais curta.
AddItem(const aText: string; aValue: Double; aColor: TsgcUIColor = 0)	Integer	Acrescenta um item, também quando o texto já está em uso. Retorna seu índice.
DeleteItem(aIndex: Integer)		Exclui um item.
ClearItems		Exclui todos os itens.
IndexOfText(const aText: string)	Integer	Índice do primeiro item com este texto (diferencia maiúsculas de minúsculas), -1 quando nenhum.
ItemCount	Integer	Número de itens.
BeginUpdate / EndUpdate		As gravações entre eles redesenham o gráfico uma única vez, no último **EndUpdate**.
BeginReload / EndReload		Mark and sweep: o último **EndReload** exclui os itens que não foram gravados desde o primeiro **BeginReload**, de modo que um recarregamento mantém as cores e os flags dos itens que permanecem. Também é uma atualização, com um único redesenho no final.

GetSnapshot(var aSnap: TsgcChartCategorySnapshot)		Copia os itens para **aSnap** sob o bloqueio quando mudaram desde a última cópia.

Os tokens de rótulo de toda série de categoria são expandidos por **sgcChartFormatLabel**, que recebe a porcentagem e a participação anterior em unidades de porcentagem (25 significa 25%) e acrescenta o sinal `%`.

Rótulos

cslInside centraliza cada rótulo em sua fatia. **cslOutside** posiciona os rótulos em duas colunas, à esquerda e à direita da pizza, ligados às suas fatias por linhas de chamada; os rótulos de uma coluna são espaçados de modo que nunca se sobreponham. **cslAuto** desenha um rótulo dentro quando ele cabe na corda e na espessura do anel, e fora caso contrário. Um rótulo mais largo que seu espaço é encurtado para caber.

Teste de acerto, dica de ferramenta e legenda

Um acerto em uma fatia preenche **Index** e **SubIndex** com o índice da coleção (**CS_CHART_OTHERS_INDEX**, -2, para a fatia Others), **YValue** com o valor, **Percent** com a participação e **Text** com o texto da fatia. A dica de ferramenta mostra `{text}: {value} ({percent})`. A legenda mostra uma entrada por fatia desenhada, Others por último. Veja TsgcChartInteraction.

Exemplo: Uma rosca com Others e uma fatia explodida

```
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.
```

Veja Também

- TsgcChartSeries
- Áreas de Plotagem
- TsgcChartInteraction
- Título, Legenda e Margens

TsgcChartSeries_Radar

Unidade: sgcCharts_Series_Radar
TsgcChartSeries_Radar desenha valores em eixos radiais (spokes) ao redor de um centro, como uma linha poligonal ou um polígono preenchido. É uma série de categoria (veja os métodos das séries de categoria no tópico da pizza): cada item de **Points** é um spoke. Um radar dá à sua área de plotagem o modo polar, e todos os radares de uma área compartilham uma única grade polar cujos spokes são a união dos textos de seus itens, na ordem do primeiro uso. A escala radial é o **YAxis** da série, de modo que **Automatic**, **Minimum**, **Maximum**, o número de marcas e o formato de rótulo desse eixo se aplicam; valores fora de seu intervalo são limitados à borda.

Propriedades

Propriedade	Tipo	Descrição
Points	TsgcChartCategoryItems	Um item por spoke: **Text** dá nome ao spoke e **Value** é o valor radial.
Filled	Boolean	Padrão False. True preenche o polígono.
FillOpacity	Byte	Alfa do preenchimento. Padrão $40.
LineWidth	Double	Largura do contorno em pixels de dispositivo na escala 1; 0 não desenha contorno.
Markers	TsgcChartSeriesMarkers	Marcadores dos vértices, visíveis por padrão.
GridShape	TsgcChartRadarGrid	Anéis `crgPolygon` (padrão) ou `crgCircle`. Usado quando este é o primeiro radar desenhado de sua área.
StartAngle	Double	Ângulo do primeiro spoke em graus, 0 (padrão) é 12 horas, no sentido horário. Usado quando este é o primeiro radar desenhado de sua área.
SpokeLabelStyle	TsgcUITextStyle	Fonte dos textos dos spokes; **Color** 0 usa a cor de texto do tema. Usado quando este é o primeiro radar desenhado de sua área.

As propriedades **GridVisible** e **GridColor** do eixo Y controlam os anéis e os spokes, e sua propriedade **Visible** alterna os rótulos das marcas. Um anel externo é acrescentado quando nenhuma marca cai na borda.

Geometria da última renderização

Método	Retorno	Descrição
CenterX, CenterY, Radius	Double (somente leitura)	Centro e raio da grade polar.
SpokeCount	Integer (somente leitura)	Número de spokes.
SpokeText[Index]	string (somente leitura)	Texto de um spoke.

VertexPoint(aSpoke: Integer; out aX, aY: Double)	Boolean	Pixel do vértice desta série no spoke **aSpoke**. False (e o centro) para um spoke inexistente, um valor que não é finito, ou quando a série não foi desenhada.

Teste de acerto e dica de ferramenta

Um acerto retorna o vértice mais próximo dos radares da área dentro do raio de clique, a partir da geometria da última renderização. **Index** é o spoke, **SubIndex** o item, **YValue** o valor, **Text** o texto do spoke e **Distance** a distância em pixels; **Percent** e **ZValue** são NaN. A dica de ferramenta mostra `Product A, Speed: 80`.

Exemplo: Dois produtos em seis critérios

```
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.
```

Veja Também

- TsgcChartSeries
- Áreas de Plotagem
- TsgcChartInteraction
- Eixos

TsgcChartSeries_Heatmap

Unidade: sgcCharts_Series_Heatmap
TsgcChartSeries_Heatmap desenha uma matriz de valores como células coloridas nos eixos X e Y. É uma série XY (descendente direta de **TsgcChartSeries**), de modo que compartilha seu gráfico com séries de linha, barra ou dispersão e usa eixos numéricos, de data e hora ou de categoria como são. Os valores ficam em um armazenamento thread-safe de linhas, **Data** (**TsgcChartHeatmapData**): com **Orientation** `chmColumns` uma linha é uma coluna em uma chave X e suas células correm ao longo de Y; com `chmRows` uma linha é uma fileira em uma chave Y. Com **Capacity** acima de 0, o armazenamento mantém apenas as linhas mais recentes, a ideia do buffer circular aplicada a uma matriz, o que transforma um heatmap em um espectrograma rolante. As cores vêm de uma escala de cores.

Propriedades

Propriedade	Tipo	Descrição
Orientation	TsgcChartHeatmapOrientation	`chmColumns` (padrão) ou `chmRows`.
Capacity	Integer	0 (padrão) deixa o armazenamento crescer; acima de 0 mantém apenas as linhas mais recentes.
CellOrigin	Double	Centro da célula 0 no eixo de células. Padrão 0.
CellStep	Double	Distância entre os centros das células, acima de 0. Padrão 1.
LineStep	Double	Extensão de uma linha no eixo de linhas; 0 (padrão) alcança a metade do caminho até as linhas vizinhas.
ColorScale	TsgcChartColorScale	Cores das células. Um **MaxColor** 0 resolve para a **Color** da série ou, na falta dela, para a primeira cor da paleta; um **MinColor** 0 para o fundo do gráfico. Veja Escala de Cores.
ShowValues	Boolean	Padrão False. True escreve o valor em toda célula pintada do caminho vetorial.
ValueFormat	string	Máscara FormatFloat dos textos de valor; vazio (padrão) mostra até 2 casas decimais.
ValueStyle	TsgcUITextStyle	Fonte dos textos de valor; **Color** 0 escolhe texto branco ou escuro pela luminância da célula.
CellBorderColor	TsgcUIColor	Borda desenhada dentro de toda célula pintada do caminho vetorial; 0 (padrão) não desenha nenhuma.
Data	TsgcChartHeatmapData (somente leitura)	O armazenamento de linhas. As gravações feitas diretamente em **Data** não redesenham o gráfico; os métodos da série abaixo redesenham.
Cells[aLine, aCell]	Double	Um valor de célula; NaN quando fora do intervalo. Uma gravação fora do intervalo é ignorada. NaN é uma célula nula, nunca pintada e nunca acertada.
ImagePath	Boolean (somente leitura)	True quando a última renderização preencheu uma imagem, False quando desenhou retângulos.

DrawnCells	Integer (somente leitura)	Células visíveis da última renderização.

Métodos

Método	Retorno	Descrição
BeginUpdate / EndUpdate		As gravações entre eles redesenham o gráfico uma única vez. Qualquer thread.
Clear		Remove todas as linhas.
SetSize(aLines, aCells: Integer)		Dimensiona a matriz para **aLines** linhas de **aCells** células.
AppendLine(const aValues: array of Double)		Acrescenta uma linha com a chave do contador de linhas em execução do armazenamento.
AppendLine(const aKey: Double; const aValues: array of Double)		Acrescenta uma linha na chave X (colunas) ou Y (fileiras) **aKey**, por exemplo um carimbo de data e hora. Uma linha mais longa que o número atual de células alarga todas as linhas e as completa com células nulas.
SetLineLabels(const aLabels: array of string) / SetLineLabels(aLabels: TStrings)		Textos das linhas, mostrados por um eixo de linhas de categoria.
SetCellLabels(const aLabels: array of string) / SetCellLabels(aLabels: TStrings)		Textos das células, mostrados por um eixo de células de categoria.

Eixos e desenho

Em um eixo de categoria, uma linha ou uma célula fica em seu índice lógico, qualquer que seja sua chave ou **CellOrigin**, de modo que um armazenamento rolante permanece alinhado com seus rótulos, que chegam ao eixo por **GetDataCategories**. Toda borda de célula se ajusta a um pixel de dispositivo inteiro. A série escolhe o caminho de desenho pelo número e pelo tamanho das células visíveis, nunca pelo canvas: até **CS_CHART_HEATMAP_VECTOR_CELLS** (20.000) células de pelo menos **CS_CHART_HEATMAP_VECTOR_MIN_SIZE** (3) pixels são desenhadas como retângulos com bordas e textos de valor; matrizes maiores são preenchidas como uma única imagem, sem bordas nem textos. A saída é a mesma nos canvases raster, SVG, VCL e FMX.

Teste de acerto, dica de ferramenta e legenda

Um acerto preenche **Area** com `chaPlot`, **Index** com a linha, **SubIndex** com a célula, **XValue** e **YValue** com o centro da célula, **ZValue** com o valor e **Text** com `line, cell` (o texto do eixo quando um rótulo está vazio); **Distance** é 0 e **Percent** é NaN. A dica de ferramenta mostra `Sales, Mar, South: 64`. Quando uma série de linha é desenhada por cima, um ponto de linha dentro do raio de clique prevalece sobre a célula. A legenda mostra uma barra de cores com o mínimo e o máximo da escala.

Exemplo: Uma matriz de região por mês

```
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.
```

Veja Também

- TsgcChartSeries
- Áreas de Plotagem
- TsgcChartInteraction
- Escala de Cores
- Eixos

TsgcChartSeries_Treemap

Unidade: sgcCharts_Series_Treemap
TsgcChartSeries_Treemap desenha uma hierarquia como retângulos aninhados cujas áreas seguem os valores. É uma série de categoria (veja os métodos das séries de categoria no tópico da pizza): o **Text** de cada item de **Nodes** é um caminho como `Europe/Spain/Madrid`, dividido por **PathDelimiter**, e os grupos são as somas de seus filhos. O layout é quadrado (squarified) por nível, com preenchimento e faixas de cabeçalho, e mantém a ordem dos irmãos sob atualizações ao vivo de valores. Um clique detalha um grupo e um breadcrumb acima dos blocos leva de volta. Um treemap é uma série de célula, veja Áreas de Plotagem.

Propriedades

Propriedade	Tipo	Descrição
Nodes	TsgcChartCategoryItems	As folhas: um caminho em **Text**, o valor em **Value**. A **Color** de um item de nível superior substitui sua cor da paleta.
PathDelimiter	Char	Separador dos segmentos do caminho. Padrão `/`.
MaxDepth	Integer	Níveis desenhados abaixo da raiz atual; abaixo de 1 não há limite. Padrão 2. Grupos mais profundos são desenhados recolhidos, como um único bloco.
ColorMode	TsgcChartTreemapColorMode	`ctcGroup` (padrão): uma cor da paleta por grupo de nível superior, clareada (tema claro) ou escurecida (tema escuro) por nível. `ctcValue`: blocos coloridos por meio de **ColorScale**, sobre o intervalo dos blocos abaixo da raiz.
ColorScale	TsgcChartColorScale	Cores de `ctcValue`; **MaxColor** 0 é a **Color** da série ou, na falta dela, a primeira cor da paleta. Veja Escala de Cores.
Levels	TsgcChartTreemapLevels	Borda, preenchimento e cabeçalho por nível; **Items[0]** é o nível 1, os filhos da raiz atual. Veja TsgcChartTreemapLevel abaixo.
LabelFormat	string	Padrão `{text}`. Os tokens são escritos como um nome entre chaves: `{text}`, `{value}` e `{percent}`; text é o último segmento do caminho e percent é a participação na raiz atual.
ValueFormat	string	Máscara FormatFloat do token de valor; vazio (padrão) mostra até 2 casas decimais.
PercentDecimals	Integer	Padrão 1.
LabelStyle	TsgcUITextStyle	Fonte dos rótulos; **Color** 0 escolhe texto branco ou escuro pelo preenchimento.
StableLayout	Boolean	Padrão True: uma atualização de valor mantém a ordem dos irmãos, de modo que os blocos não saltam com dados ao vivo.
DrillOnClick	Boolean	Padrão True: um clique em um bloco detalha seu ancestral de nível 1, um clique em uma crumb sobe até ela.
OnDrill	TNotifyEvent	Dispara quando **DrillPath** muda.
DrillPath	string (public)	Caminho da raiz atual; vazio é a raiz da árvore. Defini-lo dispara **OnDrill**.

Layout	TsgcChartTreemapLayout (somente leitura)	A árvore e a geometria da última renderização: nós, nós desenhados, o breadcrumb (**Crumbs**) e sua caixa.

TsgcChartTreemapLevel

Propriedade	Tipo	Descrição
BorderColor	TsgcUIColor	0 (padrão) usa a cor de fundo do gráfico.
BorderWidth	Double	Pixels de dispositivo na escala 1; 0 não desenha borda.
Padding	Double	Espaço entre um grupo deste nível e seus filhos, em pixels de dispositivo na escala 1.
ShowHeader	Boolean	Uma faixa de cabeçalho com o nome do grupo acima dos filhos.

Um novo item de nível assume os padrões de seu nível: 2 pixels de preenchimento e um cabeçalho para o primeiro item, 1 pixel e sem cabeçalho para os demais. Um nível sem item usa uma borda de 1 pixel.

Métodos

Método	Retorno	Descrição
DrillDown(const aPath: string)		Torna um caminho de nó a raiz atual; um caminho vazio é a raiz da árvore.
DrillUp		Torna o pai da raiz atual a raiz atual.
DrillReset		Torna a raiz da árvore a raiz atual.
NodeColor(aNode: Integer)	TsgcUIColor	Preenchimento do nó de layout **aNode** na última renderização, 0 quando não desenhado.
CrumbRect(aIndex: Integer)	TsgcUIRect	Caixa da crumb **aIndex** (na ordem de **Layout.Crumbs**) na última renderização; vazia quando não desenhada. A primeira crumb é a raiz, com o texto `All` .

Rótulos, teste de acerto e legenda

Um rótulo é exibido quando sua caixa é pelo menos tão larga quanto três caracteres e tão alta quanto uma linha; um texto mais longo é encurtado para caber, um bloco menor não mostra rótulo. Um acerto retorna o bloco mais profundo sob o ponto: **Index** é o item (-1 para um grupo, que não tem item próprio), **SubIndex** a profundidade, **YValue** o valor, **Percent** a participação na raiz atual e **Text** o caminho. Um acerto em uma crumb tem **Percent** NaN. Em `ctcGroup` a legenda lista os grupos de nível superior da árvore inteira; em `ctcValue` mostra uma barra de cores.

Exemplo: Uma árvore de três níveis com detalhamento

```
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.
```

Veja Também

- TsgcChartSeries
- Áreas de Plotagem
- TsgcChartInteraction
- Escala de Cores

TsgcChartSeries_Funnel

Unidade: sgcCharts_Series_Funnel
TsgcChartSeries_Funnel desenha as etapas de um processo, como um pipeline de vendas, como um funil que se estreita até um gargalo. É uma série de categoria (veja os métodos das séries de categoria no tópico da pizza): cada item de **Stages** é uma etapa, de cima para baixo na ordem da coleção. Um funil é uma série de célula, veja Áreas de Plotagem.

Propriedades

Propriedade	Tipo	Descrição
Stages	TsgcChartCategoryItems	As etapas, de cima para baixo.
NeckWidthPercent	Double	Largura do gargalo, porcentagem da largura do funil. Padrão 30.
NeckHeightPercent	Double	Altura do gargalo, porcentagem da altura do funil. Padrão 25.
ProportionalHeight	Boolean	False (padrão): etapas com alturas iguais. True: alturas conforme o valor.
Gap	Integer	Espaço entre duas etapas, em pixels de dispositivo na escala 1. Padrão 2.
Labels	TsgcChartSliceLabels	`cslOutside` (padrão) desenha uma coluna à direita com linhas de chamada na cor da etapa; `cslInside`, `cslAuto` e `cslNone` funcionam como na pizza.
LabelFormat	string	Padrão `{text}: {value} ({percent})`. Os tokens são escritos como um nome entre chaves: `{text}`, `{value}` e `{percent}` (participação na primeira etapa) e `{previous}` (participação na etapa acima, 100% para a primeira).
ValueFormat	string	Máscara FormatFloat do token de valor; vazio (padrão) mostra até 2 casas decimais.
PercentDecimals	Integer	Padrão 1.
LabelStyle	TsgcUITextStyle	**Color** 0 usa a cor de texto do tema fora do funil e texto branco ou escuro, escolhido pela cor da etapa, dentro dele.
BorderColor	TsgcUIColor	0 (padrão) usa a cor de fundo do gráfico.
BorderWidth	Double	Padrão 1.

Métodos e geometria da última renderização

Método	Retorno	Descrição
StageColor(aIndex: Integer)	TsgcUIColor	A cor com que a etapa **aIndex** é desenhada: sua **Color** ou, na falta dela, a cor da paleta do gráfico para **aIndex**.

HalfWidthAt(aY: Double)	Double	Meia largura da forma em **aY**; 0 acima ou abaixo do funil.
StageAt(X, Y: Double)	Integer	Posição da etapa desenhada (ou de seu rótulo externo) em **X, Y**, -1 quando nenhuma.
DrawnCount	Integer (somente leitura)	Número de etapas desenhadas.
DrawnStages[aPos]	TsgcChartFunnelStage (somente leitura)	Uma etapa desenhada, de cima para baixo: **Index** (índice da coleção), **Value**, **Percent** (da primeira etapa), **Previous** (da etapa acima), **Top**, **Bottom**, as meias larguras, o lugar e a caixa do rótulo, e os pontos da linha de chamada.
CenterX, FunnelRect, NeckTop, NeckWidth	Double, TsgcUIRect (somente leitura)	O centro horizontal, o retângulo delimitador da forma, o topo do gargalo e sua largura.

Teste de acerto, dica de ferramenta e legenda

Um acerto em uma etapa ou em seu rótulo externo preenche **Index** e **SubIndex** com o índice da coleção, **YValue** com o valor, **Percent** com a participação na primeira etapa e **Text** com o texto da etapa. A dica de ferramenta mostra `{text}: {value} ({percent})`. A legenda mostra uma entrada por etapa desenhada.

Exemplo: Um pipeline de vendas

```
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.
```

Veja Também

- TsgcChartSeries
- Áreas de Plotagem
- TsgcChartInteraction

TsgcVCLChart

Unidade: sgcCharts_VCL
TsgcVCLChart é a visão VCL de um modelo TsgcChart: um descendente de `TCustomControl`, somente Windows, que desenha o gráfico atribuído à sua propriedade **Chart**. O controle registra um listener no gráfico e se repinta sempre que o modelo muda; o modelo em si não contém código de desenho, portanto o mesmo gráfico também pode ser exibido por um TsgcFMXChart ou renderizado sem interface visual por um TsgcChartRenderer ao mesmo tempo.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este controle desenha. Atribuí-lo registra o controle como um listener; limpá-lo, ou destruir o gráfico, faz com que o controle pare de desenhar qualquer coisa.
FrameCount	Cardinal (somente leitura)	Número de quadros completos renderizados desde que o controle foi criado.
OverlayFrameCount	Cardinal (somente leitura)	Número de quadros somente de sobreposição renderizados desde que o controle foi criado. Veja **Repinturas somente de sobreposição** abaixo.

As propriedades usuais de `TCustomControl` também são publicadas sem alteração: **Align**, **Anchors**, **Color**, **Constraints**, **DragCursor**, **DragKind**, **DragMode**, **Enabled**, **ParentColor**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop**, **Visible**, e os eventos de mouse, arraste e redimensionamento.

Métodos

Método	Retorna	Descrição
SaveToPNG(const aFileName: string)		Renderiza **Chart** no tamanho de cliente e PPI atuais do controle em um novo arquivo PNG. Não faz nada quando **Chart** é nil.
SaveToSVG(const aFileName: string)		Renderiza **Chart** no tamanho de cliente e PPI atuais do controle em um arquivo SVG. Não faz nada quando **Chart** é nil.

Limitação por MaxFPS

MaxFPS não é uma propriedade deste controle, pertence ao modelo de gráfico (veja TsgcChart). Uma notificação de alteração vinda do gráfico apenas marca o controle como sujo e agenda uma repintura, no máximo **Chart.MaxFPS** vezes por segundo: quando o próximo quadro já está devido, o controle é invalidado imediatamente, caso contrário um timer de disparo único o invalida quando o quadro se tornar devido. Os quadros são ritmados contra um prazo corrente (prazo anterior mais 1000 dividido por **MaxFPS**), não contra um intervalo fixo desde a última pintura, de modo que a granularidade do timer do Windows não deixa a taxa efetiva cair abaixo do limite. Uma notificação que chega em uma thread de feed nunca toca o controle diretamente, ela envia uma mensagem privada que a thread principal recebe antes de agendar a repintura.

Repinturas somente de sobreposição

O controle também registra um segundo listener, de sobreposição, no gráfico. Uma alteração que apenas move a mira, a dica de ferramenta ou uma seleção em borracha (**TsgcChart.InvalidateOverlay**) não executa uma repintura completa: os eixos, séries, plano de fundo, título e legenda não são redesenhados. Em vez disso, os pixels da última renderização completa são copiados de volta para o raster interno e apenas **TsgcChart.RenderOverlay** é desenhado por cima, motivo pelo qual **OverlayFrameCount** e **FrameCount** são rastreados separadamente. Uma renderização completa ainda ocorre sempre que o controle fica sujo por outro motivo, como redimensionamento, uma alteração de PPI, ou uma alteração nos dados, eixos, tema ou layout do gráfico.

Exemplo: exibindo um gráfico em um formulário 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');
```

Veja Também

- TsgcChart
- TsgcFMXChart
- TsgcChartRenderer

- TsgcChartSeries_Line
- TsgcChartInteraction
- Panes

TsgcFMXChart

Unidade: sgcCharts_FMX
TsgcFMXChart é a visão FireMonkey de um modelo TsgcChart: um descendente de `TControl` que desenha o gráfico atribuído à sua propriedade **Chart** no Windows, macOS, iOS e Android. A unidade não compila nada a menos que `SGC_CHARTS_FMX` esteja definido, o que exige Delphi XE6 ou posterior. O modelo não contém código de desenho, portanto o mesmo gráfico também pode ser exibido por um TsgcVCLChart ou renderizado sem interface visual por um TsgcChartRenderer ao mesmo tempo.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este controle desenha. Atribuí-lo registra o controle como um listener; limpá-lo, ou destruir o gráfico, faz com que o controle pare de desenhar qualquer coisa.
FrameCount	Cardinal (somente leitura)	Número de quadros completos renderizados desde que o controle foi criado.
OverlayFrameCount	Cardinal (somente leitura)	Número de quadros somente de sobreposição renderizados desde que o controle foi criado. Veja **Repinturas somente de sobreposição** abaixo.

As propriedades usuais de `TControl` também são publicadas sem alteração: **Align**, **Anchors**, **ClipChildren**, **ClipParent**, **Cursor**, **DragMode**, **EnableDragHighlight**, **Enabled**, **Locked**, **Height**, **HitTest**, **Padding**, **Opacity**, **Margins**, **PopupMenu**, **Position**, **RotationAngle**, **RotationCenter**, **Scale**, **Visible**, **Width**, **TouchTargetExpansion**, e os eventos de mouse, arraste e redimensionamento.

Métodos

Método	Retorna	Descrição
SaveToPNG(const aFileName: string)		Renderiza **Chart** no tamanho e escala de cena atuais do controle em um novo arquivo PNG. Não faz nada quando **Chart** é nil.
SaveToSVG(const aFileName: string)		Renderiza **Chart** no tamanho e escala de cena atuais do controle em um arquivo SVG. Não faz nada quando **Chart** é nil.
DetectPixelOrder(var aData: TBitmapData) [class function]	TsgcFMXChartPixelOrder	Grava um pixel de teste em dados de bitmap mapeados e o lê de volta para descobrir se o bitmap da plataforma armazena a ordem de canais `fpoBGRA`, `fpoRGBA` ou outra (`fpoOther`). Usado internamente antes de copiar os pixels renderizados para um bitmap FMX; a maioria das aplicações nunca o chama diretamente.
CopyToBitmapData(aRasterizer: TsgcUIRasterizer; var aData: TBitmapData) [class procedure]		Converte os pixels ARGB diretos de **aRasterizer** nos pixels pré-multiplicados dos dados de bitmap mapeados, sobre o menor dos dois tamanhos. Usado inter-

		namente por **Paint**; a maioria das aplicações nunca o chama diretamente.

Limitação por MaxFPS

MaxFPS não é uma propriedade deste controle, pertence ao modelo de gráfico (veja TsgcChart). Uma notificação de alteração vinda do gráfico apenas marca o controle como sujo e agenda uma repintura, no máximo **Chart.MaxFPS** vezes por segundo: quando o próximo quadro já está devido, o controle é repintado imediatamente, caso contrário um timer de disparo único o repinta quando o quadro se tornar devido. Os quadros são ritmados contra um prazo corrente (prazo anterior mais 1000 dividido por **MaxFPS**), não contra um intervalo fixo desde a última pintura. Uma notificação que chega em uma thread de feed nunca toca o controle diretamente, ela enfileira uma chamada para a thread principal com **TThread.Queue** (no máximo uma chamada pendente), e a thread principal agenda a repintura; o destrutor remove uma chamada que ainda esteja enfileirada.

Repinturas somente de sobreposição

O controle também registra um segundo listener, de sobreposição, no gráfico. Uma alteração que apenas move a mira, a dica de ferramenta ou uma seleção em borracha (**TsgcChart.InvalidateOverlay**) não executa uma repintura completa: os eixos, séries, plano de fundo, título e legenda não são redesenhados. Em vez disso, os pixels da última renderização completa são copiados de volta para o raster interno e apenas **TsgcChart.RenderOverlay** é desenhado por cima, motivo pelo qual **OverlayFrameCount** e **FrameCount** são rastreados separadamente. Uma renderização completa ainda ocorre sempre que o controle fica sujo por outro motivo, como redimensionamento, uma alteração de escala de cena, ou uma alteração nos dados, eixos, tema ou layout do gráfico.

Exemplo: exibindo um gráfico em um formulário 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');
```

Veja Também

- TsgcChart
- TsgcVCLChart
- TsgcChartRenderer
- TsgcChartSeries_Line
- TsgcChartInteraction
- Panes

TsgcChartRenderer

Unidade: sgcCharts_Renderer
TsgcChartRenderer é a visão sem interface visual de um modelo TsgcChart: um `TComponent` simples sem dependência de VCL ou FMX e sem janela própria, de modo que renderiza um gráfico em um arquivo ou stream PNG, em um documento SVG ou em um buffer de pixels ARGB bruto a partir de um programa de console, um serviço do Windows, um módulo ISAPI ou um daemon Linux64. Como não possui superfície visual, não é afetado por **Chart.MaxFPS**: não há laço de quadros nem timer, cada chamada de renderização desenha o gráfico de forma síncrona no momento em que é chamada.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este renderizador desenha. Todo método Render* dispara **EsgcChartRendererError** quando este valor é nil.
Width	Integer	Largura lógica padrão em pixels usada pelas sobrecargas que não recebem tamanho. Padrão 800.
Height	Integer	Altura lógica padrão em pixels usada pelas sobrecargas que não recebem tamanho. Padrão 500.
Scale	Double	Pixels de dispositivo por pixel lógico. Padrão 1. A saída é `Round(Width * Scale)` por `Round(Height * Scale)` pixels de dispositivo; **Chart.Scale** é definido com este valor durante a renderização e restaurado ao valor anterior depois.
CompressPNG	Boolean	Se a saída PNG é comprimida com deflate. Padrão True quando a biblioteca foi compilada com `SGC_CHARTS_DEFLATE`, caso contrário padrão False.

Métodos

Método	Retorna	Descrição
RenderToPNGFile(const aFileName: string) RenderToPNGFile(const aFileName: string; aWidth, aHeight: Integer)		Renderiza para um arquivo PNG, em **Width**/**Height** ou no tamanho lógico informado, sem alterar **Width** e **Height**.
RenderToPNGStream(aStream: TStream) RenderToPNGStream(aStream: TStream; aWidth, aHeight: Integer)		Renderiza PNG (RGBA, 8 bits) para um stream.
RenderToSVG RenderToSVG(aWidth, aHeight: Integer)	string	Renderiza e retorna um documento SVG 1.1 (ASCII puro) como uma string.
RenderToSVGFile(const aFileName: string) RenderToSVGFile(const aFileName: string; aWidth, aHeight: Integer)		Renderiza SVG para um arquivo.
RenderToSVGStream(aStream: TStream) RenderToSVGStream(aStream: TStream; aWidth, aHeight: Integer)		Renderiza SVG para um stream.

RenderToBuffer(var aBuffer: TsgcUICardinalArray; out aPixelWidth, aPixelHeight: Integer) RenderToBuffer(var aBuffer: TsgcUICardinalArray; out aPixelWidth, aPixelHeight: Integer; aWidth, aHeight: Integer)		Renderiza em pixels $AARRGGBB diretos (não pré-multiplicados), de cima para baixo, com stride de linha igual a **aPixelWidth**; **aPixelWidth** e **aPixelHeight** retornam com o tamanho do dispositivo.

Concorrência

Todo método de renderização pode ser chamado de qualquer thread. Cada **TsgcChartRenderer** possui suas próprias telas, uma rasterizada e uma SVG, protegidas por um bloqueio privado daquela instância do renderizador. Além disso, toda renderização realizada por qualquer **TsgcChartRenderer** no processo passa por mais um bloqueio compartilhado por toda a unidade, porque **TsgcChart.Render** mantém o estado de layout no próprio gráfico enquanto executa. Enquanto uma renderização está em andamento, o gráfico não pode ser alterado estruturalmente, ou seja, nenhuma série ou eixo pode ser adicionado, removido ou liberado, e nenhuma propriedade que afete o layout pode ser alterada; adicionar dados por meio dos buffers das séries (**Buffer.Append**, **OHLCBuffer.Append**, as fontes de dados) é seguro, porque as séries desenham a partir de um snapshot obtido sob o bloqueio próprio do buffer.
O bloqueio de renderização compartilhado só é adquirido por **TsgcChartRenderer**. Uma visão visual, TsgcVCLChart ou TsgcFMXChart, não o adquire, portanto não renderize o mesmo gráfico com um renderizador em uma thread enquanto uma visão visual o estiver desenhando no mesmo momento.

Exemplo: renderização sem interface visual para PNG e 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;
```

Veja Também

- TsgcChart
- TsgcVCLChart
- TsgcFMXChart
- TsgcChartSeries_Line
- Panes

TsgcChartInteraction

Unidade: sgcCharts_Interaction
TsgcChartInteraction é a API de entrada de um TsgcChart: um `TPersistent`, sem dependência de framework, que transforma chamadas brutas de mouse, roda e gesto em leituras de mira, cliques em pontos, zoom, pan e, na edição financeira, comportamento de ferramenta de desenho e divisor de painel. Não carrega dependência de VCL, FMX ou de nenhuma API de janelas; uma visão repassa o que quer que receba de seus próprios eventos de entrada (um TsgcVCLChart, um TsgcFMXChart, ou um laço de host personalizado) como coordenadas de pixel do gráfico, o mesmo espaço de pixel de dispositivo do retângulo passado a **TsgcChart.Render**. Todo gráfico possui exatamente uma instância de interação, criada em seu construtor e exposta como **TsgcChart.Interaction**; as aplicações a configuram ali em vez de criar uma própria.

Propriedades

Propriedade	Tipo	Descrição
Crosshair	Boolean	Desenha a mira e as leituras de eixo enquanto o cursor passa sobre o gráfico. Padrão True.
CrosshairColor	TsgcUIColor	0 (o padrão) usa a cor do tema.
Tooltip	Boolean	Exibe a dica de ferramenta do ponto mais próximo enquanto o cursor paira sobre o gráfico. Padrão True.
ZoomMode	TsgcChartZoomMode	`czmNone`, `czmX`, `czmY` ou `czmXY`: sobre quais eixos o zoom e o pan atuam. Padrão `czmX`.
WheelZoom	Boolean	A roda do mouse aplica zoom em torno do cursor; Shift + roda faz pan horizontal em vez disso. Padrão True.
BoxZoom	Boolean	Um arraste com o botão esquerdo desenha uma seleção em borracha e aplica zoom a ela. Padrão True. Ctrl + arraste com o botão esquerdo sempre faz pan em vez disso, e um arraste com o botão direito ou do meio faz pan independentemente desta propriedade.
Pan	Boolean	Habilita o pan por arraste. Padrão True.
AutoScrollResumeDelay	Integer	Milissegundos sem interação antes que um zoom horizontal seja reiniciado e uma **TsgcChart.TimeWindow** volte a rolar; 0 significa nunca retomar automaticamente. Padrão 3000. Veja Rolagem automática.
SnapToPoint	Boolean	A mira salta para o ponto mais próximo: mais próximo em X, depois mais próximo em Y entre as séries naquele X (caso contrário, mais próximo por distância em pixels quando False). Padrão True.
DrawingTool	TsgcChartDrawingTool	Somente na edição financeira. `cdtNone` (o padrão) permite que um clique com o botão esquerdo selecione e arraste um desenho existente; qualquer outro valor (`cdtTrendLine`, `cdtHorizontalLine`, `cdtVerticalLine`, `cdtRectangle`, `cdtFibonacci`) faz com que o próximo arraste com o botão esquerdo no gráfico crie um desenho daquele tipo (pressionar = primeira âncora, soltar = segunda âncora), após o que a propriedade volta a `cdtNone`.
Paused	Boolean (somente leitura)	True enquanto a rolagem automática está pausada por um zoom ou um pan. Veja Rolagem automática.

Dragging	Boolean (somente leitura)	True enquanto um zoom por caixa, pan, divisor ou arraste de desenho está em andamento.
DragMode	TsgcChartDragMode (somente leitura)	`cdmNone`, `cdmBox`, `cdmPan`, ou, na edição financeira, `cdmSplitter` (um divisor de painel está sendo arrastado, veja Panes), `cdmDrawCreate` ou `cdmDrawMove` (um desenho ou uma de suas âncoras está sendo arrastado).
Chart	TComponent (somente leitura)	O gráfico ao qual esta interação está vinculada.

Métodos

Método	Retorna	Descrição
MouseMove(X, Y: Double)		Alimenta um movimento do ponteiro, em pixels do gráfico. Atualiza a mira, continua um arraste em andamento, ou dispara **TsgcChart.OnCrosshairMove** quando ocioso sobre o gráfico.
MouseDown(X, Y: Double; aButton: TsgcChartMouseButton; aShift: TsgcChartShiftState)		Alimenta um pressionamento de botão. **aButton** é `cmbLeft`, `cmbRight` ou `cmbMiddle`; **aShift** é um conjunto de `cskShift`, `cskCtrl`, `cskAlt`. Inicia um zoom por caixa, pan, arraste de divisor ou arraste de desenho dependendo das propriedades acima.
MouseUp(X, Y: Double; aButton: TsgcChartMouseButton; aShift: TsgcChartShiftState)		Alimenta uma liberação de botão. Completa o arraste iniciado por **MouseDown**, ou, quando o ponteiro nunca se moveu além do limiar de arraste, resolve um clique e dispara **TsgcChart.OnSeriesClick**.
MouseWheel(X, Y: Double; aDelta: Integer; aShift: TsgcChartShiftState)		Alimenta um clique de roda. **aDelta** é 120 por clique, positivo para longe do usuário (zoom para dentro).
MouseLeave		Alimenta a saída do ponteiro do controle, ocultando a mira e a dica de ferramenta.
DoubleClick(X, Y: Double)		Alimenta um clique duplo, em pixels do gráfico. Chama **ResetZoom**.
Gesture(aKind: TsgcChartGestureKind; aCenterX, aCenterY, aFactorOrDX, aDY: Double)		Alimenta um gesto de toque. Para `cgZoom`, **aFactorOrDX** é a escala do pinça (maior que 1 aplica zoom para dentro) em torno de **aCenterX, aCenterY**; para `cgPan`, **aFactorOrDX** e **aDY** são o movimento do dedo em pixels.
ResetZoom		Reinicia o zoom de todo eixo e retoma a rolagem automática.
DeleteSelectedDrawing	Boolean	Somente na edição financeira. Remove o desenho selecionado de **TsgcChart.Drawings**, a menos que esteja Locked; uma visão a chama para a tecla Delete. True quando um desenho foi removido.
HitTest(X, Y: Double)	TsgcChartHitInfo	O que está sob uma posição de pixel do gráfico após o último **TsgcChart.Render**, sem nenhum efeito colateral. Veja HitTest e TsgcChartHitInfo.

CheckAutoScroll	Boolean	Chamado por **TsgcChart.Render** antes que os intervalos sejam calculados. Reinicia o zoom horizontal assim que **AutoScrollResumeDelay** milissegundos tiverem decorrido desde a última interação; True quando isso ocorreu. Veja Rolagem automática.
DrawOverlay(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Chamado por **TsgcChart.Render** após o conteúdo do gráfico. Desenha a mira, a dica de ferramenta e a seleção em borracha dentro de **aRect** de **aCanvas**.

Eventos

Estes três eventos são publicados no próprio **TsgcChart**, não em TsgcChartInteraction: a interação é o que os dispara no gráfico ao qual está vinculada, portanto uma aplicação os atribui no gráfico. Veja TsgcChart.

Evento	Tipo	Descrição
TsgcChart.OnCrosshairMove	TsgcChartHitEvent	O ponteiro se moveu sobre o gráfico enquanto ocioso (nenhum arraste em andamento); **aHit** é o resultado de **HitTest** na posição do ponteiro.
TsgcChart.OnSeriesClick	TsgcChartHitEvent	Um clique com o botão esquerdo caiu dentro do raio de clique de um ponto; **aHit.Series** e **aHit.Index** o identificam.
TsgcChart.OnZoom	TNotifyEvent	Disparado após um zoom interativo, um pan ou um **ResetZoom**.

TsgcChartHitEvent é declarado como:

```
TsgcChartHitEvent = procedure(Sender: TObject; const aHit: TsgcChartHitInfo)
  of object;
```

HitTest e TsgcChartHitInfo

HitTest informa sobre o que uma posição de pixel do gráfico recai, usando o retângulo de plotagem calculado pelo último **TsgcChart.Render**; não altera nada e é seguro chamá-lo a qualquer momento, inclusive fora de **MouseMove**, **MouseDown** ou **MouseUp**. Ele retorna um registro **TsgcChartHitInfo** com estes campos:

Campo	Tipo	Descrição
Area	TsgcChartHitArea	A parte do gráfico sob a posição: `chaNone` , `chaPlot` , `chaAxis` , `chaLegend` ou `chaTitle` .
Axis	TsgcChartAxis	O eixo sob a posição quando **Area** é `chaAxis` .
Series	TComponent	A **TsgcChartSeries** do ponto mais próximo, nil quando nenhuma. Declarado como TComponent porque TsgcChartInteraction é compilado antes de TsgcChart e dos tipos de série.
SeriesIndex	Integer	Índice de **Series** no gráfico, -1 quando nenhum.
Index	Integer	Índice lógico do buffer do ponto, 0 é o mais antigo, -1 quando nenhum.
XValue, YValue	Double	Valores do ponto. **YValue** é o Close de um candle.

IsOHLC	Boolean	True quando o ponto vem de uma série OHLC ou candlestick.
Open, High, Low, Close	Double	Definidos apenas quando **IsOHLC** é True.
PixelX, PixelY	Double	Posição do ponto em pixels do gráfico.
Distance	Double	Distância em pixels entre a posição consultada e o ponto; 0 dentro de uma barra ou de um candle.
CursorX, CursorY	Double	A posição consultada, conforme passada a **HitTest**.
SubIndex	Integer	Item da série sob a posição: uma fatia, um ponto de radar, uma célula de heatmap, uma profundidade de treemap ou uma etapa de funil; -1 quando nenhum.
ZValue	Double	Terceiro valor do item (o valor da célula do heatmap), NaN quando nenhum.
Percent	Double	Participação do item em sua série (pie, funnel, treemap), NaN quando nenhuma.
Text	string	Texto de categoria do item (fatia, spoke, células, caminho do nó ou etapa), vazio quando nenhum.

TsgcChart.OnSeriesClick só dispara quando **Distance** está dentro de um pequeno raio de clique do ponto; um clique que cai em uma área vazia do gráfico ainda chama **HitTest** internamente, mas não dispara o evento.

Áreas de plotagem: hover, clique e ações das séries

O hover, o clique e as dicas de ferramenta alcançam as séries de todos os modos de área de plotagem: células de pie, funnel e treemap, áreas polares de radar, e o heatmap em uma área XY. O zoom, o deslocamento, a mira e a redefinição de zoom por clique duplo funcionam apenas em áreas XY. Não há novos eventos do gráfico: o hover dispara **OnCrosshairMove** e um clique dispara **OnSeriesClick**, ambos com os campos de acerto acima; depois de **OnSeriesClick**, a série executa sua própria ação de clique por meio de **AreaClick**, de modo que uma pizza com **ExplodeOnClick** puxa a fatia para fora e um treemap com **DrillOnClick** detalha o grupo. Uma série pode fornecer seu próprio texto de dica de ferramenta, que é desenhado como linhas simples. Com um heatmap sob uma série de linha, um ponto de linha dentro do raio de clique prevalece, e caso contrário prevalece a célula do heatmap.

```
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.
```

Rolagem automática

Um gráfico com uma **TsgcChart.TimeWindow** diferente de zero normalmente mantém seu eixo horizontal exibindo os dados mais recentes. Aplicar zoom ou pan nos eixos horizontais congela esse comportamento, de modo que o intervalo com zoom permaneça na tela enquanto novos dados continuam chegando; **Paused** reporta True enquanto isso está em vigor. Não há um método separado de pausa ou retomada: **TsgcChart.Render** chama **CheckAutoScroll** em cada renderização, e assim que **AutoScrollResumeDelay** milissegundos tiverem passado sem interação, o zoom horizontal é reiniciado e a janela de tempo volta a rolar; **AutoScrollResumeDelay** definido como 0 desativa a retomada automática, deixando **ResetZoom** como o único caminho de volta. Nenhum timer está envolvido, a verificação ocorre apenas no momento da renderização.

Exemplo: alimentando entrada sintética e lendo o resultado

TsgcChartInteraction recebe coordenadas simples, portanto nenhum formulário VCL ou FMX é necessário para exercitá-lo; um programa de console pode controlá-lo diretamente assim que o gráfico tiver sido renderizado ao menos uma vez (para que **PlotRect** seja conhecido):

```
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.
```

Isso imprime o resultado da mira, o resultado equivalente de **HitTest**, e a série e o índice resolvidos pelo clique, por exemplo:

```
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
```

Veja Também

- TsgcChart
- Axes
- TsgcChartSeries_Line
- TsgcVCLChart
- TsgcFMXChart
- Panes
- Áreas de Plotagem

TsgcChartDataSource

Unidade: sgcCharts_DataSource
Classe base para toda fonte de dados de gráfico. Ela insere pontos em um **Buffer** (TsgcChartBuffer) e candles em um **OHLCBuffer** (**TsgcChartOHLCBuffer**). Ambos são definidos em código e não pertencem à fonte de dados. Cada inserção passa pelo próprio bloqueio do buffer, de modo que uma thread de alimentação pode inserir enquanto a thread de renderização lê um snapshot.
Mapping informa à fonte de dados onde estão os valores X, Y e OHLC em cada registro recebido: um caminho JSON para entrada JSON, um nome de coluna ou um índice de coluna baseado em 0 para entrada CSV, ou um nome de campo para um dataset. Um **XPath** vazio faz X ser o índice do registro baseado em 0. Um registro produz um ponto quando **Buffer** está atribuído e **YPath** está definido, e um candle quando **OHLCBuffer** está atribuído e **OpenPath**, **HighPath**, **LowPath** e **ClosePath** estão definidos (**VolumePath** é opcional; um volume ausente é 0). Registros cujos X ou valores não podem ser lidos ou convertidos são ignorados.
Mapping.TimeFormat converte X: `ctfNumber` mantém o número, `ctfUnixSeconds` e `ctfUnixMilliseconds` transformam um tempo Unix em um `TDateTime` UTC, `ctfISO8601` interpreta um texto ISO 8601 (um deslocamento `Z` ou `+hh:mm` é convertido para UTC; um texto sem fuso é mantido como está) e mantém um número simples como um `TDateTime`. Os números são sempre interpretados com `.` como separador decimal, seja qual for o locale.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Active	Boolean	True	False faz a fonte de dados ignorar toda inserção.
Mapping	TsgcChartDataMapping	—	XPath, YPath, OpenPath, HighPath, LowPath, ClosePath, VolumePath, LabelPath e TimeFormat. Veja acima, e Séries de categoria e heatmaps abaixo para **LabelPath**.
Throttle	Integer	0	Milissegundos entre notificações de **OnDataChanged**. Veja abaixo.
OnBeforeAppend	TsgcChartBeforeAppendEvent	nil	Dispara antes de um ponto ser inserido, com `aX` e `aY` graváveis e `aAccept` definido como True; definir `aAccept` como False descarta o ponto.
OnBeforeAppendOHLC	TsgcChartBeforeAppendOHLCEvent	nil	Dispara antes de um candle ser inserido, com `aX`, `aOpen`, `aHigh`, `aLow`, `aClose`, `aVolume` graváveis e `aAccept` definido como True; definir `aAccept` como False descarta o candle.
OnDataChanged	TNotifyEvent	nil	Dispara depois que um ou mais pontos ou candles são inseridos. Veja Notificação abaixo.

Métodos

Método	Descrição
BeginUpdate / EndUpdate	Adia as notificações até o **EndUpdate** correspondente, que então envia uma única notificação (com throttle) quando algo foi inserido nesse meio-tempo. Aninhe livremente; apenas o **EndUpdate** mais externo pode disparar.
Clear	Limpa **Buffer** e **OHLCBuffer**, esvazia uma série de categoria ou um heatmap vinculado e reinicia o índice de registro usado quando **XPath** está vazio.

Notificação e threading

OnDataChanged tem throttle: após uma inserção, o evento dispara imediatamente quando a última notificação tem pelo menos **Throttle** ms, caso contrário uma thread auxiliar o dispara quando a janela de throttle se fecha. Nunca há mais de uma notificação pendente, de modo que uma rajada de inserções produz um único evento por janela. **Throttle** = 0 notifica a cada inserção (ainda assim combinadas enquanto uma notificação está pendente). Delphi 2010 em diante entregam **OnDataChanged** na thread principal, por meio de `TThread.Queue`; uma aplicação de console ou serviço precisa chamar `CheckSynchronize` para recebê-lo. Delphi 7 a 2009 não têm `TThread.Queue`, de modo que ali o evento dispara na thread que fez a inserção, ou na thread de throttle, e o handler precisa se sincronizar antes de tocar em um controle.

Séries de categoria e heatmaps

A propriedade **Series** vincula uma série X/Y ou de candles, e também uma série de categoria (pie, radar, funnel, treemap) ou um heatmap. **Mapping.LabelPath** é o caminho JSON ou a coluna CSV do texto do item. Um registro de categoria com rótulo chama **SetValue**(label, Y), de modo que o valor mais recente de um texto prevalece; sem um mapeamento de rótulo, define o item em X arredondado para o índice mais próximo (X é o índice do registro quando **XPath** está vazio). Um registro cujo rótulo está mapeado, mas ausente ou vazio, não grava nenhum item, e **OnBeforeAppend** continua filtrando (X, Y). Um registro de heatmap acrescenta uma linha: um registro JSON cujo **YPath** é um array (a chave é X quando **XPath** está definido, senão o contador de linhas em execução), ou as células de um registro CSV; valores que não são números são células nulas. Uma carga grava a série vinculada dentro de seu **BeginUpdate** / **EndUpdate**, de modo que ela redesenha uma única vez, e uma fonte de dados nunca grava em uma série em tempo de design, de modo que dados ao vivo nunca chegam a um DFM. As fontes dataset e tabela usam **LabelField** e **LabelColumn** em vez de **LabelPath**.

Descendentes

- TsgcChartDataSource_Manual: inserida diretamente a partir do código.
- TsgcChartDataSource_DataSet: carrega a partir de um `TDataSet`.
- TsgcChartDataSource_JSON: carrega a partir de um documento JSON.
- TsgcChartDataSource_CSV: carrega a partir de texto CSV.
- TsgcChartDataSource_Table: carrega a partir de uma tabela que implementa `IsgcUITable`, como um modelo de dados do sgcGrid.
- As fontes de dados de ponte, alimentadas por uma conexão sgcWebSockets ativa.

TsgcChartDataSource_Manual

Unidade: sgcCharts_DataSource
O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Mapping**, **Active**, **Throttle**, **BeginUpdate**/**EndUpdate**, **Clear** e os eventos de pré-inserção e de alteração de dados. Não adiciona nenhuma propriedade nova, apenas os três métodos que uma aplicação chama diretamente quando já tem os valores em mãos e não precisa que **Mapping** os localize em um registro.

Métodos

Método	Retorno	Descrição
AppendValue(const aX, aY: Double)	Boolean	Insere um ponto em **Buffer**. False quando a fonte de dados está inativa, **Buffer** é nil, ou **OnBeforeAppend** rejeitou o ponto.
AppendOHLC(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)	Boolean	Insere um candle em **OHLCBuffer**. False quando a fonte de dados está inativa, **OHLCBuffer** é nil, ou **OnBeforeAppendOHLC** rejeitou o candle.
AppendJSON(const aJSON: string)	Boolean	Lê um objeto JSON, ou um array de objetos ou arrays, por meio de **Mapping**, da mesma forma que TsgcChartDataSource_JSON lê um documento. Retorna False para JSON malformado; nunca gera exceção para isso.
AppendCategory(const aText: string; aValue: Double)	Boolean	Grava o valor de **aText** na série de categoria vinculada (um upsert; X de **OnBeforeAppend** é o índice do item). False quando a fonte de dados está inativa, nenhuma série de categoria está vinculada, ou **OnBeforeAppend** a rejeitou.
AppendLine(aKey: Double; const aValues: array of Double)	Boolean	Acrescenta uma linha em **aKey** ao heatmap vinculado. False quando a fonte de dados está inativa ou nenhum heatmap está vinculado.

Exemplo: Inserindo pontos a partir do 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;
```

Exemplo: Alimentando uma pizza e um heatmap

```
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.
```

Consulte também

- TsgcChartDataSource
- TsgcChartBuffer

TsgcChartDataSource_DataSet

Unidade: sgcCharts_DataSource_DataSet
Carrega dados de gráfico a partir das linhas de um `TDataSet`. O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Active**, **Throttle** e os eventos de pré-inserção e de alteração de dados; não usa a propriedade **Mapping** herdada, já que os campos são nomeados diretamente.
XField, **YField** e de **OpenField** a **VolumeField** nomeiam os campos. Um **XField** vazio faz X ser o índice do registro baseado em 0. Um campo de data e hora fornece seu `TDateTime` como X; um campo numérico é convertido por **Mapping.TimeFormat** (segundos ou milissegundos Unix); um campo de texto é interpretado como texto CSV. Uma linha com X ou Y nulo não gera ponto, uma linha com open, high, low ou close nulo não gera candle, e um volume nulo é 0.

Propriedades

Propriedade	Tipo	Descrição
DataSet	TDataSet	O dataset a partir do qual as linhas são carregadas.
XField	string	Campo que contém X. Vazio significa o índice do registro baseado em 0.
YField	string	Campo que contém Y.
OpenField, HighField, LowField, CloseField	string	Campos que contêm os valores OHLC. Os quatro precisam estar definidos para que uma linha gere um candle.
VolumeField	string	Campo que contém o volume. Opcional; um campo nulo ou não definido é 0.
LabelField	string	Campo de texto do item de uma série de categoria vinculada (pie, radar, funnel, treemap): cada linha chama **SetValue**(text, Y), e uma linha com texto nulo ou vazio não grava nenhum item. Sem **LabelField**, cada linha define o item em seu X (o índice do registro quando **XField** está vazio). Veja Séries de categoria e heatmaps.

Métodos

Método	Descrição
LoadFromDataSet	Insere cada linha do **DataSet** aberto, com os controles desabilitados e o registro atual restaurado ao final, e envia uma única notificação ao final. Não faz nada quando **DataSet** é nil ou está fechado.
Refresh	Limpa os buffers e chama **LoadFromDataSet** novamente. Não faz nada quando **DataSet** é nil ou está fechado.

Exemplo: Carregando um buffer a partir de um dataset

```
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;
```

Consulte também

- TsgcChartDataSource
- TsgcChartBuffer

TsgcChartDataSource_JSON

Unidade: sgcCharts_DataSource_JSON
Carrega dados de gráfico a partir de um documento JSON. O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Mapping**, **Active**, **Throttle** e os eventos de pré-inserção e de alteração de dados. **ArrayPath** seleciona os registros; um **ArrayPath** vazio é a raiz do documento. Um array de objetos ou arrays gera um registro por elemento, um objeto gera um registro. Cada registro é então lido por meio de **Mapping**, exatamente como descrito em TsgcChartDataSource. As cargas inserem nos buffers, então chame **Clear** antes para substituir os dados em vez de adicioná-los, e enviam uma única notificação ao final. JSON malformado ou um **ArrayPath** ausente geram `EsgcChartJSONError`.

Propriedades

Propriedade	Tipo	Descrição
ArrayPath	string	Caminho até o array (ou objeto) de registros dentro do documento. Vazio significa a raiz do documento.

Métodos

Método	Descrição
LoadFromString(const aJSON: string)	Carrega a partir de uma string JSON.
LoadFromStream(aStream: TStream)	Carrega a partir de um stream de texto UTF-8; um BOM é opcional.
LoadFromFile(const aFileName: string)	Carrega a partir de um arquivo UTF-8; um BOM é opcional.

Exemplo: Carregando candles a partir de um array 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;
```

Consulte também

- TsgcChartDataSource
- TsgcChartDataSource_Manual (AppendJSON)
- TsgcChartBuffer

TsgcChartDataSource_CSV

Unidade: sgcCharts_DataSource_CSV
Carrega dados de gráfico a partir de texto CSV (RFC 4180). O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Mapping**, **Active**, **Throttle** e os eventos de pré-inserção e de alteração de dados.
Os campos são separados por **Delimiter**; um campo pode ser delimitado por aspas com ", e um campo entre aspas pode conter o delimitador, quebras de linha e aspas duplicadas. As linhas terminam com CRLF, LF ou CR; linhas vazias são ignoradas. Quando **HasHeader** é True, o primeiro registro nomeia as colunas.
Os caminhos de **Mapping** selecionam as colunas: um nome de cabeçalho (sem diferenciar maiúsculas de minúsculas) ou um índice de coluna baseado em 0. Os números usam **DecimalSeparator** (. por padrão) seja qual for o locale. Uma linha cujos valores mapeados estão ausentes ou não são numéricos é ignorada. As cargas inserem nos buffers (chame **Clear** antes para substituir os dados) e enviam uma única notificação ao final.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Delimiter	Char	,	Separador de campo.
HasHeader	Boolean	True	Quando True, o primeiro registro nomeia as colunas e os caminhos de Mapping podem usar esses nomes em vez de um índice de coluna.
DecimalSeparator	Char	.	Separador decimal usado para interpretar campos numéricos, seja qual for o locale.

Métodos

Método	Descrição
LoadFromString(const aText: string)	Carrega a partir de uma string CSV.
LoadFromStream(aStream: TStream)	Carrega a partir de um stream de texto UTF-8; um BOM é opcional.
LoadFromFile(const aFileName: string)	Carrega a partir de um arquivo UTF-8; um BOM é opcional.

Exemplo: Carregando pontos a partir de um arquivo 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;
```

Consulte também

- TsgcChartDataSource
- TsgcChartBuffer

TsgcChartDataSource_Table

Unidade: sgcCharts_DataSource_Table
Carrega dados do gráfico a partir de qualquer tabela que implemente a interface `IsgcUITable` do sgcUICore (unit `sgcUICore_Table`), por exemplo o modelo de dados de uma grade sgcGrid, sem dependência entre os dois produtos. O componente descende de TsgcChartDataSource e herda **Buffer**, **Series**, **Active**, **Throttle** e os eventos antes de anexar e de alteração de dados; não usa os caminhos **Mapping** herdados, pois as colunas são nomeadas diretamente.
XColumn e **YColumn** nomeiam as colunas, comparadas sem diferenciar maiúsculas de minúsculas com os nomes das colunas da tabela (no sgcGrid, o **FieldName** da coluna). Um **XColumn** vazio faz de X o índice de linha baseado em 0. Um valor de data/hora fornece seu `TDateTime` como X, um valor numérico é convertido por **Mapping.TimeFormat** e um valor de texto é interpretado como texto CSV. Uma linha com X ou Y vazio ou não conversível não gera ponto. As linhas são lidas na ordem da tabela, que no sgcGrid é a visão ordenada e filtrada.
A fonte registra um listener na tabela. Cada alteração da tabela recarrega a tabela inteira quando **Active** é True e a versão da tabela mudou desde o último carregamento. O recarregamento é executado na thread que alterou a tabela, e a notificação do gráfico é entregue depois como nas outras fontes de dados. Uma tabela vinculada com **SetTable** deve ser desvinculada com `SetTable(nil)` antes de ser liberada; um componente **Table** é desvinculado automaticamente.

Propriedades

Propriedade	Tipo	Descrição
Table	TComponent	Um componente que implementa `IsgcUITable`. Atribuir um componente que não a implementa gera uma exceção.
XColumn	string	Coluna que contém X. Vazio significa o índice de linha baseado em 0.
YColumn	string	Coluna que contém Y.
LabelColumn	string	Coluna de texto do item de uma série de categoria vinculada, ou várias colunas separadas por `;`, cujos textos são unidos com o **PathDelimiter** de um treemap (de modo que uma tabela com as colunas Region, Country e City alimenta um treemap de três níveis) e com `/` nos demais casos; as partes finais vazias são descartadas. Para um heatmap, os rótulos das linhas.

Métodos

Método	Descrição
SetTable	Vincula qualquer objeto que implemente `IsgcUITable`, por exemplo o modelo **Data** de uma grade sgcGrid. nil desvincula.
GetTable	Retorna a tabela vinculada, nil quando nenhuma está vinculada.
LoadFromTable	Anexa todas as linhas da tabela, mantendo o bloqueio da tabela durante a leitura, e envia uma notificação no final.
Refresh	Limpa os buffers e carrega a tabela novamente. Não faz nada quando nenhuma tabela está vinculada.

Séries de categoria e heatmaps

Com uma série de categoria, cada linha chama **SetValue**(label, Y), e uma linha com rótulo vazio não grava nenhum item; sem **LabelColumn**, cada linha define o item em seu X (o índice da linha quando **XColumn** está vazio). Um recarregamento roda dentro de **BeginReload** / **EndReload** da série, de modo que os itens mantêm suas cores e flags e os itens das linhas removidas desaparecem. Com um heatmap, cada carga substitui a matriz por uma linha por registro (a chave é X quando **XColumn** está definido, senão o índice da linha). As células são as colunas de **YColumn** separadas por ; ou, quando **YColumn** está vazio, todas as colunas numéricas exceto **XColumn** e as colunas de rótulo; os rótulos das células são os nomes das colunas. Este é o caminho para o resultado de uma pivot ou de uma grade.

Exemplo: Uma pizza alimentada por uma tabela

```
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.
```

Consulte também

- TsgcChartDataSource
- TsgcChartSeries_Pie
- TsgcChartSeries_Heatmap

Visão Geral da Ponte

O pacote `sgcChartsWS` adiciona fontes de dados de gráfico alimentadas por uma conexão sgcWebSockets ativa. É um pacote separado do pacote principal `sgcCharts` e requer o sgcWebSockets para compilar; um projeto que usa apenas as fontes de dados principais (TsgcChartDataSource e suas descendentes manual, dataset, JSON e CSV) não precisa do sgcWebSockets de forma alguma.

Componentes

- TsgcChartDataSource_MQTT: exibe em gráfico as publicações MQTT recebidas por um `TsgcWSPClient_MQTT`.
- TsgcChartDataSource_WebSocket: exibe em gráfico as mensagens JSON recebidas por um `TsgcWebSocketClient`.
- TsgcChartDataSource_SSE: exibe em gráfico os Server-Sent Events recebidos por um `TsgcHTTP1Client`.
- TsgcChartDataSource_Binance: assina e exibe em gráfico um stream público de dados de mercado da Binance por meio de um `TsgcWSAPI_Binance`.

Encadeamento de handlers

Cada fonte de dados de ponte funciona da mesma forma: enquanto está **Active** e sua propriedade de cliente está atribuída, ela instala seu próprio handler de evento no cliente (`OnMQTTPublish`, `OnMessage`, `OnSSEMessage`, ou o evento de mensagem do cliente Binance) e mantém o handler que já existia antes. Cada mensagem recebida é primeiro exibida no gráfico e depois repassada a esse handler anterior, de modo que o handler da própria aplicação continua disparando exatamente como se a fonte de dados de gráfico não estivesse presente.
Definir **Active** como False, limpar a propriedade de cliente ou destruir a fonte de dados restaura o handler que existia antes, a menos que a aplicação tenha substituído o handler novamente nesse meio-tempo. Esse encadeamento ocorre apenas em tempo de execução, nunca em tempo de projeto, de modo que uma aplicação que atribui seu próprio handler depois que a fonte de dados é anexada deve atribuí-lo antes de definir a propriedade de cliente (ou antes de alternar **Active**) para manter os dois handlers.
As mensagens chegam na thread em que o cliente subjacente notifica. As inserções passam pelo bloqueio e pela notificação com throttle da classe base TsgcChartDataSource, de modo que qualquer modo de notificação é seguro de usar com uma fonte de dados de ponte.

TsgcChartDataSource_MQTT

Unidade: sgcChartsWS_DataSource_MQTT
Fonte de dados de gráfico alimentada por um cliente MQTT do sgcWebSockets. O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Mapping**, **Throttle** e os eventos de pré-inserção e de alteração de dados. Veja Visão Geral da Ponte para saber como funciona o encadeamento de handlers.
TopicFilter seleciona as publicações que são exibidas no gráfico, com os curingas MQTT `+` (um nível) e `#` (este e todo nível inferior); um **TopicFilter** vazio exibe todas as publicações.
PayloadFormat `cpfJSON` lê o payload por meio de **Mapping** (um objeto JSON ou um array de registros); `cpfNumber` interpreta o payload como um número (Y, com . como separador decimal, seja qual for o locale) e usa `Now` como X.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Active	Boolean	True	Instala ou remove o hook em **Client**.
Client	TsgcWSPClient_MQTT	nil	O cliente MQTT cujo `OnMQTTPublish` é interceptado.
PayloadFormat	TsgcChartMQTTPayloadFormat	cpfJSON	`cpfJSON` ou `cpfNumber`. Veja acima.
TopicFilter	string	''	Filtro de tópico MQTT (`+`, `#`). Vazio exibe todas as publicações.

Métodos

Método	Retorno	Descrição
ProcessPublish(const aTopic, aText: string)	Boolean	Exibe uma publicação no gráfico como se tivesse vindo de **Client**. False quando o tópico não corresponde a **TopicFilter** ou o payload não pode ser lido.

Exemplo: Exibindo leituras de sensores via MQTT

A propriedade **Client** recebe o componente de protocolo `TsgcWSPClient_MQTT`, que por sua vez está conectado ao cliente de transporte (aqui, um `TsgcWebSocketClient`) por meio de sua própria propriedade **Client**, da mesma forma que qualquer cliente MQTT do sgcWebSockets é conectado.

```
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;
```

Consulte também

- Visão Geral da Ponte
- TsgcChartDataSource

TsgcChartDataSource_WebSocket

Unidade: sgcChartsWS_DataSource_WebSocket
Fonte de dados de gráfico alimentada por um cliente WebSocket do sgcWebSockets. O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Mapping**, **Throttle** e os eventos de pré-inserção e de alteração de dados. Veja Visão Geral da Ponte para saber como funciona o encadeamento de handlers.
Cada mensagem de texto recebida pelo cliente é tratada como JSON e lida por meio de **Mapping**. **MessageFilter** é um caminho opcional que precisa existir na mensagem para que ela seja exibida no gráfico; deixe-o vazio para exibir todas as mensagens.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Active	Boolean	True	Instala ou remove o hook em **Client**.
Client	TsgcWebSocketClient	nil	O cliente WebSocket cujo `OnMessage` é interceptado.
MessageFilter	string	''	Um caminho que precisa existir na mensagem para que ela seja exibida no gráfico. Vazio exibe todas as mensagens.

Métodos

Método	Retorno	Descrição
ProcessMessage(const aText: string)	Boolean	Exibe uma mensagem no gráfico como se tivesse vindo de **Client**. False quando a mensagem não é JSON válido ou **MessageFilter** não existe nela.

Exemplo: Exibindo um feed JSON via 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;
```

Consulte também

- Visão Geral da Ponte
- TsgcChartDataSource

TsgcChartDataSource_SSE

Unidade: sgcChartsWS_DataSource_SSE
Fonte de dados de gráfico alimentada pelo stream de Server-Sent Events de um cliente HTTP do sgcWebSockets. O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Mapping**, **Throttle** e os eventos de pré-inserção e de alteração de dados. Veja Visão Geral da Ponte para saber como funciona o encadeamento de handlers.
Os dados de cada evento SSE são lidos como JSON por meio de **Mapping**. Um evento dividido em mais de um chunk do stream é armazenado em buffer internamente até estar completo.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Active	Boolean	True	Instala ou remove o hook em **Client**.
Client	TsgcHTTP1Client	nil	O cliente HTTP cujo `OnSSEMessage` é interceptado.

Métodos

Método	Retorno	Descrição
ProcessChunk(const aText: string)	Integer	Exibe no gráfico os eventos completos de um chunk do stream, como se tivessem vindo de **Client**. Um evento incompleto aguarda o próximo chunk. Retorna o número de eventos cujos dados eram JSON válido.
ResetStream	—	Descarta um evento incompleto armazenado em buffer.

Exemplo: Exibindo um feed 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;
```

Consulte também

- Visão Geral da Ponte
- TsgcChartDataSource

TsgcChartDataSource_Binance

Unidade: sgcChartsWS_DataSource_Binance
Fonte de dados de gráfico alimentada pelo cliente da API Binance do sgcWebSockets. O componente descende de TsgcChartDataSource e herda **Buffer**, **OHLCBuffer**, **Throttle** e os eventos de pré-inserção e de alteração de dados (não usa o **Mapping** herdado, já que os formatos das mensagens Binance são fixos por stream). Veja Visão Geral da Ponte para saber como funciona o encadeamento de handlers.
Stream seleciona o que é exibido no gráfico: `cbsKLine` exibe candles (em **OHLCBuffer**) construídos a partir de eventos kline/candlestick para **Symbol** e **Interval**; `cbsTicker` e `cbsTrade` exibem um valor (em **Buffer**) a partir do stream de ticker ou de trade correspondente para **Symbol**. **ClosedOnly**, quando `cbsKLine` é usado, exibe apenas os klines cujo candle já foi fechado, em vez de cada atualização em andamento.
Com `cbsTrade`, **OnTrade** informa cada trade de **Symbol** com seu horário, preço e quantidade (o campo `q`) antes de ser exibido no gráfico. Na edição financeira, atribua um TsgcChartAggregator a **Aggregator** para construir candles a partir dos trades: cada trade é então enviado para `Aggregator.AddTrade(time, price, quantity)` em vez de **Buffer**, de modo que os candles da série do agregador carregam o volume negociado, os próprios trades não são plotados como uma linha e `OnBeforeAppend` não dispara para eles. Com **Aggregator** nil, os trades são exibidos em **Buffer** como antes.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Active	Boolean	True	Instala ou remove o hook no cliente de **API**.
Aggregator	TsgcChartAggregator	nil	Edição financeira. Com `Stream = cbsTrade`, cada trade é passado para `Aggregator.AddTrade` com seu horário, preço e quantidade, em vez de ser inserido em **Buffer**. É limpo quando o agregador é liberado.
API	TsgcWSAPI_Binance	nil	O componente da API Binance cujas mensagens do cliente são interceptadas.
ClosedOnly	Boolean	False	Com `Stream = cbsKLine`, exibe no gráfico apenas os candles fechados.
Interval	TsgcWSBinanceChartIntervals	bci1m	Intervalo do kline, usado quando `Stream = cbsKLine`.
Stream	TsgcChartBinanceStream	cbsKLine	`cbsKLine`, `cbsTicker` ou `cbsTrade`.
Symbol	string	''	O símbolo Binance a ser exibido no gráfico, por exemplo `BTCUSDT`.

Métodos

Método	Retorno	Descrição
Subscribe	Integer	Assina **Stream** para **Symbol** (e **Interval** para klines). Retorna o id devolvido pela API.
Unsubscribe	Integer	Cancela a assinatura do mesmo stream. Retorna o id devolvido pela API.

ProcessMessage(const aText: string)	Boolean	Exibe uma mensagem Binance no gráfico como se tivesse vindo de **API.Client**. False quando não é uma mensagem de **Stream**/**Symbol**/**Interval**, ou quando nada foi exibido.

Eventos

Evento	Descrição
OnTrade(Sender: TObject; aTime: TDateTime; aPrice, aQty: Double)	Público. Com `Stream = cbsTrade`, dispara para cada trade de **Symbol** enquanto a fonte está ativa, antes de o trade ser inserido ou agregado: aTime é o horário do trade (UTC), aPrice é o preço e aQty é a quantidade (0 quando a mensagem não tem `q`).

Exemplo: Exibindo klines Binance ao 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;
```

Consulte também

- Visão Geral da Ponte
- TsgcChartDataSource

TsgcChartSeries_Indicator

Unidade: sgcCharts_Series_Financial
Somente nas edições Registered e Trial (excluída da edição Community; veja Edições).
TsgcChartSeries_Indicator é a base abstrata de todo indicador financeiro: uma descendente de TsgcChartSeries que lê as barras de outra série, sua **Source**, uma série candlestick ou OHLC ou uma série X/Y cujo Y representa open, high, low e close sem volume, e mantém suas próprias saídas calculadas em arrays internos em vez de desenhar os dados de origem por si mesma. Os indicadores concretos, TsgcChartSeries_SMA, TsgcChartSeries_EMA, TsgcChartSeries_WMA, TsgcChartSeries_Bollinger e TsgcChartSeries_VWAP, são plotados no mesmo eixo do preço, portanto normalmente são adicionados sobre uma série candlestick. A família dos osciladores (TsgcChartSeries_Oscillator e seus descendentes RSI e Stochastic), MACD, ATR e TsgcChartSeries_Volume também descendem de TsgcChartSeries_Indicator, mas são documentados em tópicos próprios porque normalmente são desenhados em um painel separado abaixo do preço.

Propriedades

Propriedade	Tipo	Descrição
Source	TsgcChart-Series	A série que este indicador acompanha: uma série candlestick / OHLC, ou uma série X/Y (seu Y é usado como open, high, low e close, sem volume). Alterá-la limpa as saídas e força um cálculo completo no próximo **Sync**.
LineWidth	Double	Pixels de dispositivo na escala 1. Padrão 1.5.
OutputCount	Integer (somente leitura)	Número de linhas de saída que o indicador calcula por barra, 1 para um indicador de linha única como o SMA, mais para um de várias linhas como o Bollinger.
PointCount	Integer (somente leitura)	Número de barras processadas, sendo 0 a mais antiga.
XValues[aIndex: Integer]	Double (somente leitura)	O X (geralmente o horário da barra) da barra processada em aIndex.
Values[aLine, aIndex: Integer]	Double (somente leitura)	Saída aLine (baseada em 0, até **OutputCount** - 1) da barra processada em aIndex. NaN durante o aquecimento do indicador, quando ainda não foram vistas barras suficientes para produzir um valor.

Métodos

Método	Retorno	Descrição
Sync		Coloca as saídas em dia com **Source**. Veja Sincronizando com a origem.
Recalculate		Força um cálculo completo no próximo **Sync**, por exemplo depois de reescrever barras mais antigas de **Source** no local.
GetDataRange(var aXMin, aXMax, aYMin, aYMax: Double)	Boolean	Chama **Sync** primeiro, depois retorna as extensões X e Y das saídas; False quando não há nenhuma.

Draw(aCanvas: TsgcUICanvas; aXAxis, aYAxis: TsgcChartAxis; const aPlot: TsgcUIRect; aScale: Double)		Pinta cada saída como uma linha de **EffectiveColor**. Uma descendente como o Bollinger a sobrescreve para desenhar bandas ou barras.
DrawLegendSymbol(aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aScale: Double)		Pinta o símbolo de legenda deste indicador.

Sincronizando com a origem

Um indicador nunca copia os dados de **Source**; ele lê suas barras e mantém apenas suas próprias saídas. **Sync** é o que coloca essas saídas em dia com **Source** novamente, e é chamado automaticamente por **GetDataRange** (portanto a cada renderização) e pelo hit test, além de estar disponível para ser chamado diretamente. O que ele faz depende do que mudou desde a última chamada:

- Ainda sem saídas, uma mudança de parâmetro (como definir **Period**), ou uma origem limpa ou recarregada (menos barras do que as já processadas, ou a barra processada mais recente agora tem um X diferente): um cálculo completo, reiniciando o indicador e alimentando-o novamente com cada barra, com o mesmo resultado de calculá-lo do zero.
- Barras inseridas em **Source** desde o último Sync: apenas as novas barras são alimentadas.
- A barra processada mais recente mudou no local, o caso do candle ao vivo (veja **TsgcChartSeries_OHLCBase.UpdateLastOHLC**): apenas essa última saída é recalculada.
- Barras descartadas no início de um ring buffer cheio (**BufferMode** = `cbmOverwrite` na origem): as saídas das barras descartadas são eliminadas e o cálculo continua de onde parou, de modo que as saídas restantes preservam o histórico com o qual foram calculadas.

Sync detecta qual caso se aplica comparando a quantidade de barras de **Source**, o X de sua barra mais antiga e da barra processada mais recente, e os valores OHLCV dessa barra mais recente. Chame **Recalculate** depois de reescrever barras mais antigas de **Source** no local, já que o Sync não consegue detectar esse caso sozinho. Saídas ainda em seu período de aquecimento, antes de o indicador ter visto barras suficientes para produzir um valor, são lidas como NaN por meio de **Values** e não são desenhadas. Tudo é executado na thread principal; o buffer de origem é lido sob seu próprio bloqueio.

TsgcChartSeries_SMA

Média móvel simples do close. Uma linha de saída, índice 0.

Propriedade	Tipo	Descrição
Period	Integer	Número de barras usadas na média. Padrão 20.

TsgcChartSeries_EMA

Média móvel exponencial do close, dando mais peso às barras recentes do que uma média simples. Uma linha de saída, índice 0.

Propriedade	Tipo	Descrição
Period	Integer	Número de barras das quais o fator de suavização é derivado. Padrão 20.

TsgcChartSeries_WMA

Média móvel ponderada linearmente do close, dando o maior peso à barra mais recente do período e o menor à mais antiga. Uma linha de saída, índice 0.

Propriedade	Tipo	Descrição

Period	Integer	Número de barras ponderadas. Padrão 20.

TsgcChartSeries_Bollinger

Bandas de Bollinger: uma média móvel do close com uma banda superior e uma inferior a um número escolhido de desvios padrão dela. Como ambas as bandas acompanham a mesma escala de preço da origem, o Bollinger é plotado sobre o preço como as médias móveis, não em seu próprio painel. Três linhas de saída: 0 é a banda média, 1 a banda superior, 2 a banda inferior.

Propriedade	Tipo	Descrição
Period	Integer	Número de barras sobre as quais a banda média e o desvio padrão são calculados. Padrão 20.
Deviations	Double	Número de desvios padrão que as bandas superior e inferior ficam distantes da banda média. Padrão 2.
BandFill	Boolean	Preenche a área entre a banda superior e a inferior. Padrão True.

TsgcChartSeries_VWAP

Preço médio ponderado por volume. Precisa de uma origem que carregue volume, uma série OHLC ou candlestick alimentada por meio de **AddOHLC** com um volume diferente de zero; uma origem X/Y não tem volume e calcula a média como se toda barra tivesse o mesmo peso. O X da origem é considerado o horário da barra. Uma linha de saída, índice 0.

Propriedade	Tipo	Descrição
SessionStart	TDateTime	Fração do dia em que uma sessão começa, 0 (o padrão) significando meia-noite; o cálculo reinicia a cada limite de sessão.

Exemplo: SMA, EMA e bandas de Bollinger sobre uma série candlestick

```
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;
```

Consulte também

- TsgcChartSeries_Oscillator
- TsgcChartSeries_Volume
- TsgcChartAggregator
- TsgcChartDrawing
- Painéis
- TsgcChartSeries_Candlestick

- TsgcChartSeries
- Edições

TsgcChartSeries_Oscillator

Unidade: sgcCharts_Series_Financial
Somente nas edições Registered e Trial (excluída da edição Community; veja Edições).
TsgcChartSeries_Oscillator é a base dos indicadores do estilo 0..100: uma descendente de TsgcChartSeries_Indicator que fixa suas extensões Y em 0..100 independentemente dos dados e pode desenhar um par de linhas-guia tracejadas de sobrecompra / sobrevenda. TsgcChartSeries_RSI e TsgcChartSeries_Stochastic descendem dela. **TsgcChartSeries_MACD** e **TsgcChartSeries_ATR** são documentados mais abaixo nesta mesma página porque também normalmente são desenhados em seu próprio painel, mas descendem diretamente de TsgcChartSeries_Indicator, não de TsgcChartSeries_Oscillator: sua saída é um valor derivado do preço sem um teto natural de 0..100, portanto mantêm uma faixa Y flutuante em vez de uma fixa. Os quatro indicadores normalmente são posicionados em um painel próprio abaixo do painel de preço, já que sua escala não corresponde ao preço.

Propriedades

Propriedade	Tipo	Descrição
GuideLines	Boolean	Desenha as linhas-guia de sobrecompra e sobrevenda como linhas horizontais tracejadas. Padrão True.

Overbought e Oversold não são publicadas na própria classe base; cada oscilador concreto as publica com seu próprio nível padrão, já que RSI e Stochastic não concordam sobre onde esses níveis ficam. Veja TsgcChartSeries_RSI e TsgcChartSeries_Stochastic abaixo.
Todo membro de TsgcChartSeries_Indicator, incluindo **Source**, as propriedades do estilo **Period** nas classes concretas, **Sync**, **PointCount** e **Values**, continua se aplicando aqui; veja Sincronizando com a origem para saber como um indicador se mantém em dia com sua série **Source**.

TsgcChartSeries_RSI

Índice de força relativa do close. Uma linha de saída, índice 0. As linhas-guia usam por padrão 30 e 70.

Propriedade	Tipo	Descrição
Period	Integer	Número de barras sobre as quais o ganho e a perda médios são calculados. Padrão 14.
Overbought	Double	Nível da linha-guia. Padrão 70.
Oversold	Double	Nível da linha-guia. Padrão 30.

TsgcChartSeries_Stochastic

Oscilador estocástico. Duas linhas de saída: 0 é %K, 1 é %D (uma média móvel de %K). As linhas-guia usam por padrão 20 e 80.

Propriedade	Tipo	Descrição
KPeriod	Integer	Período de retrospecto para %K. Padrão 14.
KSmoothing	Integer	Período de suavização aplicado a %K. Padrão 3.

DPeriod	Integer	Período da média móvel de %K que produz %D. Padrão 3.
DColor	TsgcUI-Color	Cor da linha %D; %K é desenhada em **EffectiveColor** como um indicador de linha única. Padrão para uma cor de destaque laranja.
Overbought	Double	Nível da linha-guia. Padrão 80.
Oversold	Double	Nível da linha-guia. Padrão 20.

TsgcChartSeries_MACD

Convergência / divergência de médias móveis do close. Descende diretamente de TsgcChartSeries_Indicator: sua escala acompanha as diferenças de preço a partir das quais é construída, portanto não tem uma faixa fixa de 0..100 nem linhas-guia de sobrecompra / sobrevenda. Ela sobrescreve a faixa Y para que a linha de base zero sobre a qual o histograma se estende esteja sempre visível. Três linhas de saída: 0 é a linha MACD, 1 a linha de sinal, 2 o histograma (linha MACD menos linha de sinal).

Propriedade	Tipo	Descrição
FastPeriod	Integer	Período da EMA rápida. Padrão 12.
SlowPeriod	Integer	Período da EMA lenta. Padrão 26.
SignalPeriod	Integer	Período da EMA da linha MACD que produz a linha de sinal. Padrão 9.
SignalColor	TsgcUI-Color	Cor da linha de sinal; a linha MACD usa **EffectiveColor** e o histograma é desenhado como barras de alta / baixa. Padrão para uma cor de destaque laranja.

TsgcChartSeries_ATR

Amplitude média verdadeira (average true range), uma medida de volatilidade. Descende diretamente de TsgcChartSeries_Indicator pelo mesmo motivo que o MACD: seu valor é uma faixa de preço positiva sem limite, não uma razão 0..100, portanto mantém uma faixa Y flutuante e não tem linhas-guia. Uma linha de saída, índice 0.

Propriedade	Tipo	Descrição
Period	Integer	Número de barras usadas na média. Padrão 14.

Exemplo: RSI e MACD em seus próprios painéis

```
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;
```

Consulte também

- TsgcChartSeries_Indicator

- TsgcChartSeries_Volume
- Painéis
- TsgcChartSeries_Candlestick
- TsgcChartSeries
- Edições

TsgcChartSeries_Volume

Unidade: sgcCharts_Series_Financial
Somente edições Registered e Trial (excluída da edição Community; veja Edições).
TsgcChartSeries_Volume desenha as barras de volume de uma **Source** OHLC, coloridas conforme aquela barra tenha fechado em alta ou em baixa. Ela descende diretamente de TsgcChartSeries_Indicator (não de TsgcChartSeries_Oscillator): o volume não tem teto fixo, portanto mantém uma faixa Y flutuante com sua linha de base fixada em zero, em vez de uma escala 0..100. Ela calcula duas saídas por barra, 0 é o volume e 1 é a direção (1 para fechamento >= abertura, 0 caso contrário), mas apenas a saída 0 conta para os extremos de Y, de modo que a marca de direção nunca estica o eixo. Assim como RSI, Stochastic, MACD e ATR, ela normalmente é colocada em seu próprio painel abaixo do preço, já que sua escala não corresponde à do preço. Veja Sincronizando com a fonte para saber como ela mantém suas saídas alinhadas com **Source**; as mesmas regras se aplicam aqui.

Propriedades

Propriedade	Tipo	Descrição
UpColor	TsgcUI-Color	Cor de uma barra cujo fechamento é maior ou igual à sua abertura. Por padrão, usa a cor de alta do gráfico com opacidade reduzida.
DownColor	TsgcUI-Color	Cor de uma barra cujo fechamento é menor que sua abertura. Por padrão, usa a cor de baixa do gráfico com opacidade reduzida.
BarWidth-Percent	Integer	Parcela do slot X ocupada por uma barra, 1 a 100. Padrão 70.

Todo membro de TsgcChartSeries_Indicator também se aplica: **Source**, **Sync**, **Recalculate**, **PointCount**, **XValues** e **Values**. Leia o volume da barra processada em aIndex com `Values[0, aIndex]` e sua marca de alta/baixa com `Values[1, aIndex]`.

Métodos

Método	Re-tor-na	Descrição
Draw(aCanvas: TsgcUICanvas; aXAxis, aYAxis: TsgcChartAxis; const aPlot: Tsg-cUIRect; aScale: Double)		Sobrescrito para desenhar barras de volume, barras de alta em **UpColor** e barras de baixa em **DownCo-lor**, em vez da linha que TsgcChartSeries_Indicator desenha por padrão.
DrawLegendSymbol(aCanvas: TsgcUI-Canvas; const aRect: TsgcUIRect; aScale: Double)		Sobrescrito para corresponder ao símbolo de legenda de barra em vez do símbolo de linha.

Exemplo: barras de volume abaixo de uma série candlestick

```
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;
```

Veja Também

- TsgcChartSeries_Indicator
- TsgcChartSeries_Oscillator
- Panes
- TsgcChartSeries_Candlestick
- TsgcChartSeries
- Edições

Painéis

Unidade: sgcCharts_Chart
Somente nas edições Registered e Trial (excluída da edição Community; veja Edições).
Um TsgcChart normalmente tem uma área de plotagem. Quando **Panes** contém uma ou mais entradas, essa área de plotagem é dividida em faixas horizontais empilhadas de cima para baixo, cada uma com seu próprio eixo vertical e sua própria série, enquanto cada painel continua compartilhando os eixos horizontais do gráfico. Este é o layout que um gráfico de preços com RSI, MACD ou volume abaixo dele precisa: os candles de preço em um painel, um oscilador em outro abaixo, ambos rolando e aplicando zoom juntos porque compartilham o mesmo eixo inferior. **TsgcChartPane** é uma faixa da coleção, **TsgcChartPanes** é a própria coleção, e um eixo se junta a um painel por meio de **TsgcChartAxis.Pane**.

Propriedades de TsgcChartPane

Propriedade	Tipo	Descrição
Name	string	Procurado por **TsgcChartAxis.Pane** e por **TsgcChartPanes.Find** (sem diferenciar maiúsculas de minúsculas).
HeightPercent	Double	Peso do painel: os painéis visíveis dividem a altura da plotagem proporcionalmente a esse valor, não como uma porcentagem literal que precisa somar 100. Padrão 100.
Visible	Boolean	Um painel oculto não ocupa altura de plotagem e seus eixos e séries não são desenhados. Padrão True.

Métodos de TsgcChartPanes

Método	Retorno	Descrição
Add	TsgcChartPane	Adiciona um painel ao final da pilha.
Find(const aName: string)	TsgcChartPane	Primeiro painel com esse Name (sem diferenciar maiúsculas de minúsculas), nil quando nenhum.

Propriedades e métodos de TsgcChart

Membro	Tipo	Descrição
Panes	TsgcChartPanes (propriedade)	Vazio (o padrão) significa uma área de plotagem; uma coleção não vazia divide a plotagem entre os painéis visíveis, de cima para baixo na ordem da coleção.
HasPanes	Boolean (método)	True quando o último **Render** dividiu a plotagem em painéis.
PaneRect(aIndex: Integer)	TsgcUIRect (método)	Retângulo do painel neste índice da coleção no último **Render**, vazio quando o painel está oculto.
AxisPlotRect(aAxis: TsgcChartAxis)	TsgcUIRect (método)	Retângulo de plotagem usado por aAxis no último **Render**: o painel de um eixo vertical quando o gráfico tem painéis (vazio quando seu painel está oculto), senão o **PlotRect** inteiro.

SeriesPlotRect(aSeries: TsgcChartSeries)	TsgcUIRect (método)	Retângulo de plotagem do painel de aSeries, o painel do seu eixo vertical.
SplitterAt(X, Y: Double)	Integer (método)	Índice do painel acima do divisor sob X, Y (pixels do gráfico, o espaço entre dois painéis visíveis), -1 quando o ponto não está sobre um divisor.
MoveSplitter(aIndex: Integer; Y: Double)	(método)	Move o divisor abaixo do painel em aIndex para Y (pixels do gráfico): altera o **HeightPercent** desse painel e do próximo painel visível, mantendo a soma dos dois igual.

Vinculando um eixo e uma série a um painel

TsgcChartAxis.Pane (uma string, vazia por padrão) nomeia o painel ao qual um eixo vertical pertence; um **Pane** vazio recorre ao primeiro painel visível. Um eixo horizontal ignora **Pane** por completo, já que todo painel compartilha os mesmos eixos horizontais. Uma série não tem uma propriedade **Pane** própria: ela é desenhada no painel do seu **YAxis**, de modo que posicionar uma série em um painel é inteiramente uma questão de qual eixo vertical seu **AxisY** nomeia, e o **Pane** desse eixo. Um gráfico com um painel de preço e um painel de RSI abaixo dele normalmente tem três eixos, um eixo inferior compartilhado e um eixo esquerdo por painel, cada eixo esquerdo com seu **Pane** definido para o **Name** do painel correspondente.

Arrastando o divisor

Os painéis não são redimensionados automaticamente: um host que permite ao usuário arrastar o espaço entre dois painéis chama **SplitterAt** em um mouse down para descobrir se o ponteiro está sobre um divisor (e qual painel fica acima dele), e então chama **MoveSplitter** com esse índice de painel a cada movimento do mouse durante o arraste, passando o novo Y em pixels do gráfico. **MoveSplitter** mantém constante a soma do **HeightPercent** dos dois painéis em cada lado do divisor, de modo que os painéis mais abaixo na pilha não são afetados. É exatamente assim que TsgcChartInteraction conduz o arraste do divisor a partir dos eventos de mouse VCL e FMX; um código que renderiza sem uma view e nunca recebe entrada de mouse não tem motivo para chamar nenhum dos dois métodos.

Exemplo: Um painel de preço e um painel de RSI compartilhando o eixo 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;
```

Consulte também

- TsgcChartSeries_Indicator
- TsgcChartSeries_Oscillator
- TsgcChartSeries_Volume
- TsgcChartInteraction
- Eixos
- TsgcChart
- Edições

TsgcChartAggregator

Unidade: sgcCharts_Aggregator
Somente nas edições Registered e Trial (excluída da edição Community; veja Edições).
TsgcChartAggregator é um `TComponent` que constrói os candles de **Series**, uma série candlestick ou OHLC, a partir de um stream de trades: chame **AddTrade(aTime, aPrice, aQty)** para cada trade, sendo aTime um `TDateTime` UTC. Cada candle cobre um bucket de **Interval** e seu X é o início desse bucket (o horário de abertura, a mesma convenção de um kline da Binance). Um trade no bucket do candle em formação atualiza seu high, low, close e volume no local; um trade em um bucket posterior fecha esse candle e insere um novo, aberto no início do seu próprio bucket, com **FillGaps** inserindo opcionalmente um candle plano para cada bucket vazio entre eles; um trade mais antigo que o candle em formação é ignorado e contabilizado em **DroppedCount**. O primeiro trade após a criação, ou após **Reset**, continua o candle mais recente de **Series** quando esse candle cai no mesmo bucket (histórico já carregado antes de os trades ao vivo começarem), e é descartado quando o candle mais recente pertence a um bucket posterior.
Os buckets são calculados em milissegundos inteiros, alinhados em UTC: segundos, minutos e horas se alinham ao relógio, dias às 00:00, semanas na segunda-feira às 00:00 e meses no primeiro dia do mês do calendário. **SessionAlign** desloca cada limite por um número de minutos, para sessões que não começam às 00:00 UTC.

Propriedades

Propriedade	Tipo	Descrição
Series	TsgcChartSeries_OHLCBase	A série de candles que recebe os candles. Alterá-la, ou a liberação da série, esquece o candle em formação da mesma forma que **Reset**.
Interval	TsgcChartAggregatorInterval	Tamanho do bucket: `cai1s`, `cai5s`, `cai15s`, `cai30s`, `cai1m`, `cai3m`, `cai5m`, `cai15m`, `cai30m`, `cai1h`, `cai2h`, `cai4h`, `cai6h`, `cai12h`, `cai1d`, `cai1w` ou `cai1Month`. Padrão `cai1m`. Alterá-lo esquece o candle em formação.
SessionAlign	Integer	Minutos adicionados a cada limite UTC, 0 (o padrão) significando limites UTC simples. Alterá-lo esquece o candle em formação.
FillGaps	Boolean	Insere um candle plano (open = high = low = close = o close anterior, volume 0) para cada bucket vazio entre dois trades. Padrão False.
DroppedCount	Integer (somente leitura)	Número de trades ignorados por serem mais antigos que o bucket do candle em formação.

Métodos

Método	Retorno	Descrição
AddTrade(aTime: TDateTime; aPrice, aQty: Double)		Alimenta um trade. Atualiza **Series** da mesma forma que **AddOHLC** e **UpdateLastOHLC** fariam, portanto precisa ser chamado a partir da mesma thread em que esses são chamados.
Reset		Esquece o candle em formação e limpa **DroppedCount**. A própria **Series** não é alterada.

BucketStart(aTime: TDateTime)	TDateTime	Início do bucket em que aTime cai, respeitando **Interval** e **SessionAlign**.

Alimentando o agregador

Trades da Binance são agregados por um TsgcChartDataSource_Binance em quatro passos: coloque uma série candlestick ou OHLC no gráfico (uma TsgcChartSeries_Volume em seu próprio painel pode exibir o volume dos candles); atribua essa série a **Series** e escolha **Interval**; defina o `Stream` da fonte de dados como `cbsTrade`; atribua o agregador à sua propriedade `Aggregator`. Cada trade então chega a **AddTrade** com seu horário, preço e quantidade, de modo que os candles carregam o volume negociado, e os trades não são plotados como uma linha. Qualquer outro feed chama **AddTrade** por conta própria, a partir de qualquer thread que produza os trades, desde que seja a mesma thread a partir da qual **Series** é atualizada de outra forma.

Exemplo: Agregando trades em candles de um 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;
```

Consulte também

- TsgcChartSeries_Indicator
- TsgcChartDrawing
- Painéis
- Visão Geral da Ponte
- TsgcChartDataSource_Binance
- TsgcChartSeries_Candlestick
- Edições

TsgcChartDrawing

Unidade: sgcCharts_Drawing
Somente nas edições Registered e Trial (excluída da edição Community; veja Edições).
TsgcChart.Drawings contém itens **TsgcChartDrawing**: linhas de tendência, linhas horizontais e verticais, retângulos e retrações de Fibonacci, ancorados em valores de dados (**X1**, **Y1**, **X2**, **Y2**) dos eixos nomeados por **AxisX** e **AxisY**, de modo que um desenho acompanha o zoom, o pan e os dados ao vivo em vez de permanecer em uma posição fixa de pixel. **TsgcChart.RenderContent** pinta os desenhos depois das séries e dos eixos, recortados (clipped) ao painel do seu eixo vertical. Além dos desenhos, uma série carrega seus próprios marcadores de preço, **TsgcChartSeries.LastValueMarker** e **TsgcChartSeries.BidAskBand**, ambos `TPersistent` e ambos documentados nesta página.

Propriedades de TsgcChartDrawing

Propriedade	Tipo	Descrição
Kind	TsgcChartDrawingKind	`cdkTrendLine` (o segmento entre as duas âncoras), `cdkHorizontalLine` (uma linha através do painel em Y1), `cdkVerticalLine` (uma linha através do painel em X1), `cdkRectangle` (a caixa das duas âncoras, contorno mais um preenchimento leve) ou `cdkFibonacci` (os níveis de retração entre X1 e X2). Padrão `cdkTrendLine`.
X1, Y1, X2, Y2	Double	Âncoras, como valores de dados de **AxisX** (X) e **AxisY** (Y). Para `cdkFibonacci`, 0% fica em Y2 e 100% em Y1 (arraste do início do movimento até o fim); os níveis abrangem toda a largura do painel quando X1 é igual a X2.
AxisX	string	Vazio significa o primeiro eixo inferior, a mesma convenção de uma série.
AxisY	string	Vazio significa o primeiro eixo esquerdo; o desenho é pintado no painel deste eixo. Veja Painéis.
Color	TsgcUIColor	0 (o padrão) significa a primeira cor da paleta do gráfico.
LineWidth	Double	Pixels de dispositivo na escala 1. Padrão 1.5.
Dash	TsgcChartDrawingDash	`cddSolid`, `cddDash` ou `cddDot`. Padrão `cddSolid`.
Visible	Boolean	Padrão True.
Locked	Boolean	Um desenho bloqueado ainda pode ser selecionado, mas não movido ou excluído interativamente. Padrão False.

Método de TsgcChartDrawing

Método	Retorno	Descrição
SetAnchors(const aX1, aY1, aX2, aY2: Double)		Define as quatro âncoras com uma única notificação de alteração, em vez de uma por propriedade.

TsgcChartDrawings (TsgcChart.Drawings)

TsgcChartDrawings é a `TOwnedCollection` por trás de **TsgcChart.Drawings**.

Membro	Retorno	Descrição
Add	Tsgc-Chart-Drawing	Adiciona um desenho.
AnchorPixels(aIndex: Integer; out aX1, aY1, aX2, aY2: Double; out aPlot: TsgcUIRect)	Boolean	Âncoras do desenho em aIndex, em pixels do gráfico, conforme o último **Render**; também retorna o retângulo do painel em que o desenho foi pintado.
PixelToValue(aIndex: Integer; X, Y: Double; out aX, aY: Double)	Boolean	Valores de dados do pixel do gráfico X, Y nos eixos do desenho em aIndex, conforme o último **Render**.
HitTest(X, Y: Double; out aHandle: Integer)	Integer	Índice do desenho sob X, Y (pixels do gráfico), o de cima primeiro, -1 quando nenhum. aHandle é a âncora sob o ponto, 0 ou 1, ou -1 para um acerto no corpo.
DeleteSelected	Boolean	Remove **Selected**, a menos que esteja **Locked**; True quando foi removido.
Draw(aCanvas: TsgcUICanvas; aScale: Double)		Pinta todo desenho visível. Chamado por **TsgcChart.RenderContent**.
DrawSelection(aCanvas: TsgcUICanvas; aScale: Double)		Pinta as alças de **Selected**, parte da camada de interação.
SaveToJSON	string	Um objeto com um array "drawings", um objeto por desenho: "kind" (trendline, hline, vline, rectangle ou fibonacci), "x1", "y1", "x2", "y2", "axisX", "axisY", "color" (ARGB como um número sem sinal), "lineWidth", "dash" (solid, dash ou dot), "visible" e "locked". Os números usam '.' independentemente do locale e preservam cada dígito do double.
LoadFromJSON(const aJSON: string)		Substitui os desenhos (um array na raiz também é aceito); um item de um tipo desconhecido é ignorado, e JSON malformado gera **EsgcChartJSONError**.
Selected	Integer (propriedade)	Índice do desenho selecionado, -1 quando nenhum.

Desenho e seleção interativos

TsgcChartInteraction.DrawingTool cria um desenho com o mouse: pressionar define a primeira âncora, arrastar define a segunda, soltar o finaliza e a ferramenta volta para `cdtNone`. Sem nenhuma ferramenta ativa, um clique seleciona o desenho sob o ponteiro (**Drawings.Selected**), um arraste sobre seu corpo o move, um arraste sobre uma alça move apenas aquela âncora, e **TsgcChartInteraction.DeleteSelectedDrawing**, vinculado à tecla Delete nas views VCL e FMX, o remove por meio de **DeleteSelected**. Veja TsgcChartInteraction para a conexão de mouse e teclado; um código sem view chama **Add**, **HitTest** e **Selected** diretamente.

TsgcChartLastValueMarker

Uma linha tracejada através do painel no último valor de uma série, com uma caixa de rótulo para esse valor sobre o eixo vertical, do lado da plotagem.

Propriedade	Tipo	Descrição
Visible	Boolean	Padrão False.
Color	TsgcUICo-lor	0 (o padrão) significa a cor da série; para uma série de candles, a cor de alta ou de baixa do último candle.

TsgcChartBidAskBand

Bid e Ask, definidos por código, desenhados como uma faixa clara entre os dois preços, uma linha em cada um e uma caixa de rótulo para cada um sobre o eixo vertical.

Propriedade	Tipo	Descrição
Visible	Boolean	Padrão False.
Bid, Ask	Double	Definidos em conjunto com **SetPrices** para gerar uma única notificação de alteração.
BidColor, AskColor	TsgcUI-Color	Padrão para uma cor de destaque verde (bid) e vermelha (ask).

SetPrices(const aBid, aAsk: Double) define os dois preços com uma única notificação, em vez de duas.
Ambos os marcadores, assim como os desenhos, são pintados para uma série X/Y (seu último Y) e para uma série de candles (seu último close); uma série pode exibir um, ambos, ou nenhum, independentemente das outras séries no mesmo gráfico.

Exemplo: Uma linha de tendência, o marcador de último valor e uma faixa de 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;
```

Consulte também

- TsgcChartSeries_Indicator
- TsgcChartAggregator
- Painéis
- TsgcChartInteraction

- TsgcChartSeries_Candlestick
- TsgcChartSeries
- Edições

TsgcChartBuffer e TsgcChartOHLCBuffer

Unidade: sgcCharts_Buffer
Buffers circulares para séries de gráfico. **TsgcChartBuffer** mantém pontos X/Y, **TsgcChartOHLCBuffer** mantém X, Open, High, Low, Close e Volume. X é um `Double` (um `TDateTime` é armazenado como um `Double`). Append é O(1). No modo `cbmOverwrite`, um buffer cheio descarta seu ponto mais antigo na próxima adição, no modo `cbmGrow` a capacidade dobra.
Todo método público adquire o bloqueio do buffer, de modo que uma thread de feed pode adicionar dados enquanto a thread de renderização lê. A thread de renderização deve chamar **Snapshot** ou **SnapshotRange**, que copiam os dados para um registro de propriedade do chamador sob o bloqueio e o liberam imediatamente, e então decimar e desenhar a partir dessa cópia; veja Decimação.
Espera-se que X seja monotônico não decrescente. Append mantém a flag **Monotonic**: ela se torna False assim que um ponto chega com um X menor que o anterior, e apenas **Clear** a reinicia. Quando **Monotonic** é True, **SnapshotRange** encontra o intervalo visível por busca binária e adiciona o ponto mais próximo fora de cada borda, de modo que as linhas que saem do gráfico sejam desenhadas. Quando é False, **SnapshotRange** recorre a uma varredura linear que copia apenas os pontos dentro do intervalo, e as funções de decimação retornam os dados sem decimação.

Propriedades e métodos compartilhados (TsgcChartBufferBase)

Membro	Descrição
Capacity: Integer	Capacidade atual em pontos ou candles. Aumentá-la realoca; reduzi-la abaixo de **Count** descarta os dados mais antigos.
Count: Integer	Número de pontos ou candles atualmente mantidos.
Mode: TsgcChartBufferMode	`cbmOverwrite` (descarta o ponto mais antigo quando cheio) ou `cbmGrow` (dobra a capacidade quando cheio).
Monotonic: Boolean	True enquanto todo X adicionado tiver sido >= ao X anterior. Veja acima.
Lock / Unlock	Adquire ou libera o bloqueio próprio do buffer, para um chamador que precisa que mais de uma chamada seja atômica.
Clear	Esvazia o buffer e reinicia **Monotonic** para True.

TsgcChartBuffer

Membro	Descrição
Create(aCapacity: Integer = 1024; aMode: TsgcChartBufferMode = cbmOverwrite)	Construtor.
Append(const aX, aY: Double)	Adiciona um ponto.
AppendBatch(const aX, aY: array of Double)	Adiciona vários pontos em uma única chamada bloqueada.
First(out aX, aY: Double): Boolean	Ponto mais antigo. False quando o buffer está vazio.

Last(out aX, aY: Double): Boolean	Ponto mais novo. False quando o buffer está vazio.
Snapshot(var aSnap: TsgcChartBufferSnapshot)	Copia todo ponto para `aSnap`. Os arrays são reutilizados entre chamadas e podem ser mais longos que `aSnap.Count`.
SnapshotRange(const aXMin, aXMax: Double; var aSnap: TsgcChartBufferSnapshot)	Copia os pontos visíveis em [aXMin, aXMax], mais os pontos de borda descritos acima quando **Monotonic**.
X[aIndex], Y[aIndex]: Double	Ordem lógica: 0 é o ponto mais antigo, `Count - 1` o mais novo.

TsgcChartOHLCBuffer

Membro	Descrição
Create(aCapacity: Integer = 1024; aMode: TsgcChartBufferMode = cbmOverwrite)	Construtor.
Append(const aX, aOpen, aHigh, aLow, aClose: Double; const aVolume: Double = 0)	Adiciona um candle.
AppendBatch(const aX, aOpen, aHigh, aLow, aClose, aVolume: array of Double)	Adiciona vários candles em uma única chamada bloqueada. `aVolume` pode ficar vazio, o volume então é 0.
First / Last(out aX, aOpen, aHigh, aLow, aClose, aVolume: Double): Boolean	Candle mais antigo / mais novo. False quando o buffer está vazio.
UpdateLast(const aX, aOpen, aHigh, aLow, aClose, aVolume: Double): Boolean	Substitui os valores do candle mais novo quando seu X é igual a `aX`, para um candle ao vivo ainda em formação. Retorna False e não altera nada quando o buffer está vazio ou o candle mais novo tem outro X.
Snapshot(var aSnap: TsgcChartOHLCSnapshot)	Copia todo candle para `aSnap`.
SnapshotRange(const aXMin, aXMax: Double; var aSnap: TsgcChartOHLCSnapshot)	Copia os candles visíveis em [aXMin, aXMax].
X[aIndex], Open[aIndex], High[aIndex], Low[aIndex], Close[aIndex], Volume[aIndex]: Double	Ordem lógica: 0 é o candle mais antigo, `Count - 1` o mais novo.

Exemplo: adicionando dados e obtendo um snapshot

```
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;
```

Veja Também

- Decimação
- TsgcChartDataSource

Decimação

Unidade: sgcCharts_Buffer
Funções sem dependência de framework que reduzem um snapshot de buffer a aproximadamente um conjunto de pontos por coluna de pixel, para um gráfico com `aPixelWidth` pixels de largura exibindo `[aXMin, aXMax]`. Um snapshot não monotônico, um intervalo vazio, ou `aPixelWidth < 1` é copiado sem decimação em todos os casos abaixo.

Valor / Função	Descrição
TsgcChartDecimation = (dmNone, dmMinMax, dmLTTB)	`dmNone` copia todo ponto. `dmMinMax` divide X em colunas de pixel e mantém, para cada coluna, o primeiro, o mínimo, o máximo e o último ponto em ordem de X (no máximo 4), de modo que a polilinha permanece visualmente exata; pontos à esquerda de `aXMin` compartilham uma coluna extra, pontos à direita de `aXMax` outra. `dmLTTB` retorna `2 × aPixelWidth` pontos escolhidos por Largest Triangle Three Buckets, destinado a séries estáticas.
sgcChartDecimate(const aSnap: TsgcChartBufferSnapshot; const aXMin, aXMax: Double; aPixelWidth: Integer; aMode: TsgcChartDecimation; var aOutX, aOutY: TsgcChartDoubleArray; out aOutCount: Integer)	Decima um snapshot de pontos usando `aMode`. `aOutX` e `aOutY` são ampliados quando curtos demais, nunca reduzidos.
sgcChartLTTB(const aX, aY: TsgcChartDoubleArray; aCount, aThreshold: Integer; var aOutX, aOutY: TsgcChartDoubleArray; out aOutCount: Integer)	Largest Triangle Three Buckets diretamente em um par de arrays X/Y. Retorna exatamente `Min(aThreshold, aCount)` pontos, sempre incluindo o primeiro e o último.
sgcChartDecimateOHLC(const aSnap: TsgcChartOHLCSnapshot; const aXMin, aXMax: Double; aPixelWidth: Integer; var aOut: TsgcChartOHLCSnapshot)	Mescla os candles que caem na mesma coluna de pixel: X e abertura do primeiro, máxima mais alta, mínima mais baixa, fechamento do último, soma dos volumes. `aOut.Count` mantém o resultado.

Exemplo: decimando um snapshot para um gráfico 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;
```

Veja Também

- TsgcChartBuffer e TsgcChartOHLCBuffer

TsgcUICanvas

Unidade: sgcUICore_Canvas, com dois backends: sgcUICore_Canvas_Raster (**TsgcUICanvas_Raster**) e sgcUICore_Canvas_SVG (**TsgcUICanvas_SVG**)
TsgcUICanvas é a superfície de desenho abstrata em que TsgcChart.**Render** desenha. O pacote principal distribui dois backends concretos: **TsgcUICanvas_Raster** desenha em um buffer ARGB em memória e o salva como PNG, **TsgcUICanvas_SVG** constrói um documento SVG. Ambos compartilham a mesma API de caminho, pintura e texto, de modo que um gráfico é renderizado de forma idêntica com qualquer um deles. Uma aplicação normalmente apenas cria a tela, chama **Render**, e salva o resultado; as primitivas de desenho abaixo também são públicas para adicionar sobreposições personalizadas.

Tipos

Tipo	Descrição
TsgcUIColor = Cardinal	`$AARRGGBB` direto (não pré-multiplicado).
TsgcUIRect	`Left`, `Top`, `Right`, `Bottom`: Double, pixels de dispositivo; Right / Bottom são as bordas distantes.
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	Descrição
Clear(aColor: TsgcUIColor)	Preenche toda a superfície.
BeginPath, MoveTo, LineTo, QuadTo, CubicTo, Arc, RoundRect, Rectangle, ClosePath	Construção de caminho. **Arc** recebe ângulos em radianos, 0 em +X, varredura positiva no sentido horário na tela.
FillPath(aColor: TsgcUIColor; aRule: TsgcUIFillRule = frNonZero)	Preenche o caminho atual; mantido até o próximo **BeginPath**.
FillPathLinearGradient(aX1, aY1, aX2, aY2: Double; const aStops: array of TsgcUIGradientStop; aRule: TsgcUIFillRule = frNonZero)	Preenche o caminho atual com um gradiente linear.
StrokePath(aColor, aWidth[, aJoin, aCap[, aDash, aDashOffset]])	Duas sobrecargas: com ou sem um padrão de traço.
DrawText, MeasureText, TextWidth, LineHeight, FitText	Desenho e medição de texto com um **TsgcUIFontStyle**. Veja Fontes.
DrawImage(const aImage: TsgcUIImage; aX, aY: Integer; aOpacity: Double = 1)	Copia uma imagem ARGB.

PushClip(aLeft, aTop, aRight, aBottom: Integer), PopClip	Retângulo de recorte em pixels de dispositivo, Right / Bottom exclusivos; recortes aninhados se intersectam.
SetTransform(aA, aB, aC, aD, aE, aF: Double), ResetTransform	Transformação afim: `x' = A*x + C*y + E`, `y' = B*x + D*y + F`.
SetFontFile(const aRegular, aBold: string), GetFont(aBold: Boolean)	Fontes usadas para medir e desenhar texto; um nome de arquivo vazio seleciona o Roboto embutido. Veja Fontes.
Width, Height	Tamanho da superfície em pixels de dispositivo.

Adicionais de TsgcUICanvas_Raster

Membro	Descrição
Create(aWidth, aHeight: Integer)	Sobrecarga que aloca seu próprio rasterizador.
Create(aRasterizer: TsgcUIRasterizer)	Sobrecarga que desenha em um rasterizador existente.
SetSize(aWidth, aHeight: Integer)	Redimensiona a superfície.
SaveToPNGStream(aStream: TStream; aCompress: Boolean = True)	Codifica a superfície como PNG.
SaveToPNGFile(const aFileName: string; aCompress: Boolean = True)	Codifica e grava o PNG em disco.
Rasterizer	TsgcUIRasterizer (somente leitura), o rasterizador subjacente.

Adicionais de TsgcUICanvas_SVG

Membro	Descrição
Create(aWidth, aHeight: Integer)	Cria um documento vazio do tamanho informado.
SetSize(aWidth, aHeight: Integer)	Novo documento vazio do tamanho informado.
SaveToString	string, o documento completo com a declaração XML; recortes abertos são fechados na saída.
SaveToStream(aStream: TStream), SaveToFile(const aFileName: string)	Grava o documento.

Renderizando um gráfico para PNG ou SVG

Crie a tela no tamanho de destino, chame **TsgcChart.Render** com um **TsgcUIRect** cobrindo-a, então salve. Este é todo o caminho de código de que um relatório sem interface visual ou uma exportação agendada precisa, sem nenhum controle VCL ou FireMonkey envolvido.

Exemplo: o mesmo gráfico renderizado para PNG e para 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;
```

Veja Também

- TsgcChart
- Fontes

TsgcUIFont

Unidade: sgcUICore_Font
TsgcUIFont é o mecanismo de texto TrueType em Pascal puro que toda TsgcUICanvas usa para medir e desenhar texto; ele interpreta o próprio arquivo de fonte, sem dependência da pilha de texto do sistema operacional. Toda tela cria seu próprio **TsgcUIFont** regular e negrito por meio de **SetFontFile** / **GetFont**, portanto uma aplicação raramente cria um diretamente, exceto para medir texto fora de uma tela.

Propriedades

Propriedade	Tipo	Descrição
Loaded	Boolean (somente leitura)	True assim que um arquivo ou stream de fonte foi interpretado.
UnitsPerEm, Ascender, Descender, LineGap	Integer (somente leitura)	Métricas de fonte em unidades de fonte.
GlyphCount	Integer (somente leitura)	
CmapFormat	Integer (somente leitura)	O formato da tabela de caractere para glifo encontrado no arquivo.
HasKerning	Boolean (somente leitura)	True apenas quando o arquivo de fonte carregado tem, ele mesmo, uma tabela kern.
Kerning	Boolean	Padrão True. Aplica a tabela kern da fonte, quando ela tem uma, durante a medição e o desenho; defina como False para desativar o kerning mesmo em uma fonte que tenha uma tabela.
CacheCapacity, CacheCount	Integer	Tamanho e contagem atual de entradas do cache interno de contornos de glifo.

Métodos

Método	Retorna	Descrição
LoadFromFile(const aFileName: string)		Interpreta um arquivo de fonte TrueType (contorno glyf). Dispara **EsgcUIFont** para uma fonte CFF (contorno PostScript), ou qualquer outra coisa que não seja uma fonte TrueType utilizável.
LoadFromStream(aStream: TStream)		O mesmo, a partir de um stream.
LoadDefault(aBold: Boolean = False)		Carrega a fonte embutida no sgcCharts (Roboto Regular ou Bold). Sempre funciona; nenhum arquivo é necessário.
LoadPlatformDefault(aBold: Boolean = False)		Melhor esforço para a fonte de UI do sistema para a plataforma atual (Segoe UI no Windows, Roboto no Android, DejaVu Sans ou Liberation Sans no Linux, Helvetica ou Arial no macOS/iOS); recorre a **LoadDefault** quando ne-

		nhum desses arquivos é encontrado, portanto nunca dispara uma exceção.
CharToGlyph(aCode: Cardinal)	Integer	Ponto de código Unicode para id de glifo; 0 (.notdef) quando não mapeado.
GlyphAdvance(aGlyph: Integer)	Integer	Largura de avanço em unidades de fonte.
GetKerning(aLeft, aRight: Integer)	Integer	Ajuste da tabela kern em unidades de fonte para um par de glifos; 0 sem um par.
IsCompositeGlyph(aGlyph: Integer)	Boolean	
GetGlyphOutline(aGlyph: Integer)	TsgcUIGlyphOutline	Contorno em cache em unidades de fonte, pertencente à fonte, válido até a próxima chamada que possa descartá-lo; nil para um id de glifo inválido.
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 caber em **aMaxWidth**.
DrawText(aRaster: TsgcUIRasterizer; const aText: string; aX, aY: Double; aSize: Double; aColor: Cardinal; aAngle: Double = 0)		Desenha diretamente em um rasterizador, abaixo do nível de **TsgcUICanvas**.
TextToCodePoints(const aText: string) (class function)	TsgcUICodePointArray	

Fonte embutida e fontes de plataforma

O sgcCharts embute uma compilação TrueType da Roboto para que todo gráfico desenhe texto sem nenhum arquivo externo: **LoadDefault** a carrega diretamente, e é o que um nome de arquivo vazio passado a **TsgcUICanvas.SetFontFile** seleciona. **LoadPlatformDefault** tenta primeiro a própria fonte de UI da plataforma e só recorre à embutida quando não consegue encontrar ou interpretar um arquivo utilizável, portanto é a opção padrão mais segura para uma aplicação que deseja corresponder ao sistema operacional hospedeiro. A compilação embutida da Roboto não carrega tabela kern, portanto **HasKerning** reporta False para ela, ainda que **Kerning** permaneça True; uma fonte TrueType personalizada carregada com sua própria tabela kern recebe kerning automaticamente.

Somente contornos TrueType são suportados

TsgcUIFont lê a tabela de contorno `glyf` diretamente; não possui interpretador CFF (contorno PostScript). Muitas fontes de sistema que parecem simples arquivos TrueType, como Helvetica ou SF no macOS, são na verdade CFF, e **LoadFromFile** / **LoadFromStream** disparam **EsgcUIFont** para elas. Onde uma fonte padrão de plataforma se revela CFF, **LoadPlatformDefault** detecta isso e passa para o próximo candidato, terminando na fonte embutida caso nada utilizável seja encontrado.

Exemplo: fonte embutida, fonte de plataforma e medição 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;
```

Veja Também

- Tela

Indicadores

Unidade: sgcCharts_Indicators *(somente na edição financeira, compilada sob* `SGC_CHARTS_FINANCIAL`*; um shell vazio caso contrário)*
Calculadoras de indicadores técnicos incrementais, sem dependência de framework: **TsgcChartIndicatorSMA**, **EMA**, **WMA**, **Bollinger**, **RSI**, **MACD**, **Stochastic**, **ATR** e **VWAP**, todas descendendo de **TsgcChartIndicatorCalc**. Não mantêm dependência de TsgcChart e podem alimentar uma série extra, um mecanismo de sinais, ou código de relatório simples.

Padrão comum

Membro	Descrição
Add(...)	Alimenta uma nova barra e retorna a saída mais recente. O(1) amortizado, exceto Stochastic, que varre sua janela de máxima/mínima (O(KPeriod)).
UpdateLast(...)	Substitui a entrada da barra adicionada mais recentemente, para um candle ao vivo ainda em formação: o estado salvo antes do último **Add** é restaurado e o novo valor é adicionado novamente, de modo que o resultado corresponde a um cálculo do zero com a barra modificada. Em uma calculadora vazia, comporta-se como **Add**.
Calculate(const aInput: array of Double; ...)	Reinicia a calculadora e alimenta um array inteiro por meio de **Add**, de modo que os resultados em lote e incrementais coincidem. A calculadora mantém seu estado depois, de modo que barras ao vivo podem ser adicionadas após um carregamento de histórico.
Reset	Limpa todo o estado; **Count** volta a 0.
Count	Integer (somente leitura). Barras adicionadas desde o último **Reset**.

Aquecimento

Toda saída é `NaN` (`Math.NaN`, testada com `Math.IsNan`) até que o período de aquecimento da calculadora termine; uma entrada `NaN` é ignorada, portanto não entra na janela e a saída para aquela barra também é `NaN`. Os índices de barra abaixo começam em 0.

Definições

Calculadora	Construtor	Definição
TsgcChartIndicatorSMA	Create(aPeriod: Integer = 20)	Média aritmética dos últimos **Period** valores; primeiro valor na barra Period-1.
TsgcChartIndicatorEMA	Create(aPeriod: Integer = 20)	alpha = 2/(Period+1), semeada com a SMA dos primeiros Period valores na barra Period-1, então Value := X * alpha + Prev * (1 - alpha).
TsgcChartIndicatorWMA	Create(aPeriod: Integer = 20)	Média móvel ponderada linearmente, pesos 1..Period (Period no mais novo), dividida por Period(Period+1)/2; primeiro valor na barra Period-1.

TsgcChartIndicatorBollinger	Create(aPeriod: Integer = 20; const aDeviations: Double = 2)	Middle = SMA(Period), desvio padrão populacional (dividido por Period), Upper / Lower = Middle +/- Deviations * StdDev; primeiro valor na barra Period-1.
TsgcChartIndicatorRSI	Create(aPeriod: Integer = 14)	Suavização de Wilder. Change(i) = Close(i) - Close(i-1); o primeiro ganho/perda médio é a média das primeiras Period alterações (barra Period), então Avg := (Avg * (Period-1) + Current) / Period. RSI = 100 - 100 / (1 + AvgGain / AvgLoss); 100 quando AvgLoss é 0, 50 quando ambos são 0.
TsgcChartIndicatorMACD	Create(aFastPeriod: Integer = 12; aSlowPeriod: Integer = 26; aSignalPeriod: Integer = 9)	Line = EMA(Fast) - EMA(Slow), primeira Line na barra Slow-1. Signal = EMA(Signal) da Line, semeada com a SMA dos primeiros valores de Signal Line, primeira na 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 as últimas máximas/mínimas de KPeriod (50 quando HighestHigh = LowestLow), primeiro na barra KPeriod-1. K = SMA(KSmoothing) de Raw (KSmoothing = 1 é o estocástico rápido), D = SMA(DPeriod) de K.
TsgcChartIndicatorATR	Create(aPeriod: Integer = 14)	Wilder. TR(i) = Max(High-Low, |High-PrevClose|, |Low-PrevClose|) a partir da barra 1. O primeiro ATR é a média das primeiras Period TRs (barra Period), então ATR := (ATR * (Period-1) + TR) / Period.
TsgcChartIndicatorVWAP	Create(const aSessionStart: TDateTime = 0)	Cumulativo Sum(TypicalPrice * Volume) / Sum(Volume), TypicalPrice = (High+Low+Close)/3. As somas reiniciam na primeira barra de cada sessão; a sessão de uma barra é Floor(Time - SessionStart), portanto SessionStart = 0 reinicia a cada dia do calendário e 9.5/24 reinicia às 09:30. Sem aquecimento; NaN enquanto o volume da sessão for 0.

Exemplo: SMA e RSI, em streaming e em lote

```
{$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}
```

Veja Também

- TsgcChartSeries

Visão Geral do .NET

O **sgcCharts .NET** é uma porta gerenciada em C# da biblioteca sgcCharts do Delphi, não um wrapper em torno dela: cada unidade Delphi tem uma contraparte em C# sob o namespace **esegece.sgcCharts**, usando os mesmos nomes de classe **Tsgc***, de modo que o código escrito contra a API Delphi se lê da mesma forma em C#. A porta está dividida em seis pacotes NuGet, de modo que um projeto traz apenas a plataforma que ele visa.

Pacotes

PackageId	Contém	Target Frameworks
esegece.sgcCharts	O modelo de gráfico principal: **TsgcChart**, eixos, painéis, séries, fontes de dados, buffers, desenhos, os auxiliares de matemática/formatação e JSON, e a tela SVG. Sem dependência nativa.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcCharts.Rendering	**TsgcUICanvas_Raster**, a tela rasterizada do SkiaSharp usada para saída PNG sem interface visual no Windows, Linux e macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcCharts.WinForms	**TsgcWinFormsChart**, a visão Windows Forms.	net472, net8.0-windows, net9.0-windows
esegece.sgcCharts.Wpf	**TsgcWpfChart**, a visão WPF.	net472, net8.0-windows, net9.0-windows
esegece.sgcCharts.Avalonia	**TsgcAvaloniaChart**, a visão multiplataforma (Windows, Linux, macOS).	net8.0, net9.0
esegece.sgcCharts.sgcWebSockets	A ponte ao vivo: fontes de dados de gráfico alimentadas por uma conexão **sgcWebSockets** (mensagens WebSocket, publicações MQTT, server-sent events, streams de mercado da Binance). Depende do pacote NuGet **esegece.sgcWebSockets**.	netstandard2.0, net8.0, net9.0

Edições Community e Registered

A compilação padrão de cada pacote acima (sem opção alguma) é a edição paga **Registered**: sem constante `SGC_COMMUNITY`, sem logotipo embutido, sem aviso de inicialização. Compilar ou empacotar com `-p:SgcCommunity=true` produz, em vez disso, a edição **Community** (gratuita), adicionando um sufixo **.Community** ao id de pacote de cada pacote, exceto a ponte (**esegece.sgcCharts.sgcWebSockets**), que a edição Community não distribui de forma alguma, a mesma regra que a edição Community do Delphi segue para suas próprias unidades de ponte. A compilação Community mantém todas as oito séries, candlestick e OHLC incluídas, e todas as visões, e descarta apenas o pacote financeiro (painéis, séries de indicador e de volume, agregador, ferramentas de desenho, marcador de último valor e faixa de compra/venda).
A edição Community é gratuita para uso comercial e não comercial por equipes pequenas: indivíduos e organizações com receita bruta anual abaixo de EUR 1.000.000 e no máximo 5 desenvolvedores. Organizações que atinjam ou ultrapassem qualquer um dos limites precisam de uma licença Comercial (Registered).

A API espelha o Delphi

As classes de visão mantêm a nomenclatura do Delphi: **TsgcWinFormsChart** e **TsgcWpfChart** são as contrapartes .NET do **TsgcVCLChart** do Delphi, e **TsgcAvaloniaChart** é a contraparte do **TsgcFMXChart** do Delphi.
Notificações no modelo principal, como **TsgcChart.OnChange**, **TsgcChartDataSource.OnDataChanged**, **TsgcChartDataSource.OnBeforeAppend**, **TsgcChart.OnCrosshairMove**, **OnSeriesClick** e **OnZoom**, são simples propriedades delegadas `get; set;`, não membros `event` do C#. Atribuir um novo handler substitui o anterior, exatamente como um único ponteiro de método do Delphi. O único lugar em que um evento multicast real do C# (`+=` / `-=`) está envolvido é onde a ponte **esegece.sgcCharts.sgcWebSockets** se inscreve em um cliente da porta .NET do sgcWebSockets; veja Visão Geral da Bridge e Diferenças do .NET em Relação ao Delphi.
A maioria das propriedades de array indexadas do Delphi se torna indexadores reais do C#: **TsgcChartAxes**, **TsgcChartPanes**, **TsgcChartDrawings** e **TsgcChartPalette** expõem todas `this[int Index]`, e **TsgcChartJSONNode** expõe `this[int aIndex]` para seus elementos de array. Os buffers de pontos por trás de uma série são a exceção: **TsgcChartBuffer.X(Index)** / **Y(Index)**, **TsgcChartOHLCBuffer.X(Index)** / **Open(Index)** / **High(Index)** / **Low(Index)** / **Close(Index)** / **Volume(Index)**, e os **XValues(Index)** / **Values(Line, Index)** das séries financeiras são métodos comuns, porque o C# permite apenas uma assinatura de indexador por classe e cada uma dessas precisa de vários acessores por índice, nomeados independentemente ou com múltiplos parâmetros, na mesma classe. Veja Diferenças do .NET em Relação ao Delphi.

Visões e telas descartáveis

Classes que mantêm uma superfície Skia nativa persistente ou um bitmap GDI dimensionado para o controle implementam `IDisposable` e devem ser descartadas, ou sua memória nativa vaza:

- **TsgcUICanvas_Raster** (esegece.sgcCharts.Rendering): possui o bitmap, a tela, os paints e os tipos de fonte do SkiaSharp.
- **TsgcChartRenderer**: possui sua tela rasterizada.
- **TsgcWinFormsChart**: descarta sua tela rasterizada, seu bitmap Skia e seu bitmap GDI+, e seu timer de limitação, a partir de um `Dispose(bool)` sobrescrito.
- **TsgcWpfChart**: descarta sua tela rasterizada e seu bitmap Skia.
- O quadro interno de **TsgcAvaloniaChart** (**TsgcAvaloniaChartFrame**): descarta suas imagens Skia dianteira e traseira e suas telas rasterizadas.

O próprio modelo de gráfico principal (**TsgcChart**, séries, fontes de dados) não mantém nenhum recurso nativo e não precisa de descarte por esse motivo; ainda assim implementa `IDisposable` em alguns pontos para executar uma notificação de liberação no estilo Delphi, como desvincular uma série de sua fonte ou uma fonte de dados de seu cliente.

Métricas de texto idênticas ao Delphi

Toda tela mede o texto com **TsgcUIFont**, um leitor TrueType portado da parte de MÉTRICAS do `sgcUICore_Font.pas` do Delphi: ele lê uma tabela `cmap` formato 4 ou 12 para a busca de caractere para glifo, larguras de avanço a partir de `hmtx`, pares de kerning a partir de uma subtabela `kern` formato 0, e o ascender, descender e o espaçamento entre linhas a partir de `hhea`. Ele carrega os mesmos arquivos embutidos **Roboto-Regular.ttf** e **Roboto-Bold.ttf** que a biblioteca Delphi embute (Apache License 2.0), de modo que um layout calculado no .NET corresponde ao layout do Delphi até o último bit. O SkiaSharp é usado apenas para rasterizar os contornos de glifo que o leitor de métricas já posicionou, um glifo posicionado por caractere, com antialiasing e posicionamento subpixel e sem hinting; ele nunca mede o texto por si só, o que é o que mantém todo backend, tanto raster quanto SVG, em sintonia com o Delphi.

Veja Também

- Diferenças do .NET em Relação ao Delphi
- TsgcChart
- TsgcVCLChart
- Visão Geral da Bridge

Início Rápido: WinForms

TsgcWinFormsChart é a visão Windows Forms de um modelo TsgcChart: um descendente de `Control` que desenha o gráfico atribuído à sua propriedade **Chart**. É distribuído no pacote **esegece.sgcCharts.WinForms**, destinado a `net472`, `net8.0-windows` e `net9.0-windows`, e carrega uma `ProjectReference` (ou uma `PackageReference`, quando consumido a partir do NuGet) para **esegece.sgcCharts.Rendering** de forma transitiva, portanto um projeto WinForms só precisa referenciar o próprio **esegece.sgcCharts.WinForms**. O controle é desenhado pela mesma tela rasterizada do Skia que o TsgcVCLChart do Delphi, diretamente em uma seção DIB do GDI persistente, limitada por **Chart.MaxFPS**, e é seguro alimentá-lo com dados a partir de uma thread em segundo plano.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este controle desenha. Atribuí-lo registra o controle como um listener; limpá-lo, ou descartar o gráfico, faz com que o controle pare de desenhar qualquer coisa.
FrameCount	uint (somente leitura)	Número de quadros completos renderizados desde que o controle foi criado.
OverlayFrameCount	uint (somente leitura)	Número de quadros somente de sobreposição renderizados desde que o controle foi criado, para uma alteração que apenas move a mira, a dica de ferramenta ou uma seleção em borracha.

As propriedades usuais de `Control` também são públicas e permanecem sem alteração: **Dock**, **Anchor**, **BackColor**, **Enabled**, **Visible**, **TabIndex**, **TabStop**, **ContextMenuStrip**, e os eventos de mouse, arraste e redimensionamento.

Métodos

Método	Retorna	Descrição
SaveToPNG(string aFileName)	void	Renderiza **Chart** no tamanho de cliente e DPI atuais do controle em um novo arquivo PNG. Não faz nada quando **Chart** é null.
SaveToSVG(string aFileName)	void	Renderiza **Chart** no tamanho de cliente e DPI atuais do controle em um arquivo SVG. Não faz nada quando **Chart** é null.

Exemplo: exibindo um gráfico em um formulário WinForms

Veja Também

- Visão Geral do .NET
- Início Rápido: WPF
- TsgcVCLChart
- TsgcChartSeries_Line

Início Rápido: WPF

TsgcWpfChart é a visão WPF de um modelo TsgcChart: um descendente de `Control` que desenha o gráfico atribuído à sua propriedade de dependência **Chart**. É distribuído no pacote **esegece.sgcCharts.Wpf**, destinado a `net472`, `net8.0-windows` e `net9.0-windows`, e carrega uma `ProjectReference` (ou uma `PackageReference`, quando consumido a partir do NuGet) para **esegece.sgcCharts.Rendering** de forma transitiva, portanto um projeto WPF só precisa referenciar o próprio **esegece.sgcCharts.Wpf**. O controle é desenhado pela mesma tela rasterizada do Skia que o TsgcVCLChart do Delphi, diretamente no back buffer de um `WriteableBitmap` persistente, limitada por **Chart.MaxFPS**, e é seguro alimentá-lo com dados a partir de uma thread em segundo plano.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este controle desenha, apoiado pela propriedade de dependência **ChartProperty**. Atribuí-lo registra o controle como um listener enquanto ele está carregado; limpá-lo, ou descartar o gráfico, faz com que o controle pare de desenhar qualquer coisa.
FrameCount	long (somente leitura)	Número de quadros completos renderizados desde que o controle foi criado.
OverlayFrameCount	long (somente leitura)	Número de quadros somente de sobreposição renderizados desde que o controle foi criado, para uma alteração que apenas move a mira, a dica de ferramenta ou uma seleção em borracha.

As propriedades usuais de `Control` também são públicas e permanecem sem alteração: **Width**, **Height**, **HorizontalAlignment**, **VerticalAlignment**, **Background**, **IsEnabled**, **Visibility**, **Focusable**, e os eventos roteados de mouse e teclado.

Métodos

Método	Retorna	Descrição
SaveToPNG(string aFileName)	void	Renderiza **Chart** no tamanho de renderização e DPI atuais do controle em um novo arquivo PNG. Não faz nada quando **Chart** é null.
SaveToSVG(string aFileName)	void	Renderiza **Chart** no tamanho de renderização e DPI atuais do controle em um arquivo SVG. Não faz nada quando **Chart** é null.

Exemplo: exibindo um gráfico em uma janela WPF

Veja Também

- Visão Geral do .NET
- Início Rápido: WinForms
- TsgcVCLChart
- TsgcChartSeries_Line

Início Rápido: Avalonia

Pacote: esegece.sgcCharts.Avalonia (destinado a net8.0 e net9.0, depende de **esegece.sgcCharts.Rendering**)
TsgcAvaloniaChart é a visão Avalonia de um modelo **TsgcChart**: um descendente de `Control` do Avalonia que desenha o gráfico atribuído à sua propriedade **Chart**, no Windows, Linux e macOS. É a contraparte .NET do TsgcFMXChart do Delphi. O controle registra um listener no gráfico enquanto está anexado a uma árvore visual e se repinta sempre que o modelo muda, desenhado com SkiaSharp em um bitmap de quadro persistente e exibido por meio do lease de tela Skia do renderizador Avalonia. O modelo em si não contém código de desenho, portanto o mesmo gráfico também pode ser exibido por uma visão WinForms ou WPF, ou renderizado sem interface visual, ao mesmo tempo.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este controle desenha. Atribuí-lo registra o controle como um listener enquanto está anexado a uma janela; limpá-lo, ou descartar o gráfico, faz com que o controle pare de desenhar qualquer coisa.
FrameCount	uint (somente leitura)	Número de quadros completos renderizados desde que o controle foi criado.
OverlayFrameCount	uint (somente leitura)	Número de quadros somente de sobreposição renderizados desde que o controle foi criado (uma alteração de mira, dica de ferramenta ou seleção em borracha que reaproveita o último conteúdo em cache em vez de uma repintura completa).

As propriedades usuais de `Control` do Avalonia também estão disponíveis sem alteração: **Width**, **Height**, **Margin**, **HorizontalAlignment**, **VerticalAlignment**, **Focusable**, e os eventos de ponteiro, roda e teclado.

Métodos

Método	Retorna	Descrição
SaveToPNG(string aFileName)		Renderiza **Chart** no tamanho e DPI atuais do controle em um novo arquivo PNG. Não faz nada quando **Chart** é null.
SaveToSVG(string aFileName)		Renderiza **Chart** no tamanho e DPI atuais do controle em um arquivo SVG. Não faz nada quando **Chart** é null.

O ritmo de quadros segue **Chart.MaxFPS** da mesma forma que as demais visões visuais: uma notificação de alteração apenas marca o controle como sujo e agenda uma repintura, no máximo **Chart.MaxFPS** vezes por segundo, de modo que uma thread em segundo plano pode alimentar a série do gráfico sem inundar a thread de UI com pinturas.

Exemplo

Uma aplicação Avalonia mínima que exibe um **TsgcAvaloniaChart** vinculado a um **TsgcChart** com uma série de linha, construída inteiramente em código (sem marcação XAML necessária). Além de **esegece.sgcCharts.Avalonia**, o projeto também precisa dos pacotes NuGet **Avalonia.Desktop** e **Avalonia.Themes.Fluent**, os mesmos de qualquer outra aplicação desktop Avalonia:

Veja Também

- Visão Geral do .NET
- Início Rápido: WinForms
- Início Rápido: WPF
- TsgcFMXChart (Delphi)

Renderização Sem Interface Visual

Pacote: esegece.sgcCharts.Rendering (destinado a netstandard2.0, net472, net6.0, net8.0 e net9.0)
TsgcChartRenderer é a visão sem interface visual de um modelo **TsgcChart**: uma classe simples sem dependência de nenhum framework de UI e sem janela própria, de modo que renderiza um gráfico em um arquivo ou stream PNG, em um documento SVG ou em um buffer de pixels ARGB direto a partir de um programa de console, um serviço ou uma aplicação ASP.NET. Funciona no Windows, Linux e macOS, incluindo um contêiner Linux ou um job de CI sem display conectado, e é a contraparte .NET do TsgcChartRenderer do Delphi. Como não possui superfície visual, não é afetado por **Chart.MaxFPS**: não há laço de quadros nem timer, cada chamada de renderização desenha o gráfico de forma síncrona no momento em que é chamada.

Propriedades

Propriedade	Tipo	Descrição
Chart	TsgcChart	O modelo de gráfico que este renderizador desenha. Todo método Render* dispara **EsgcChartRendererError** quando este valor é null.
Width	int	Largura lógica padrão em pixels usada pelas sobrecargas que não recebem tamanho. Padrão 800.
Height	int	Altura lógica padrão em pixels usada pelas sobrecargas que não recebem tamanho. Padrão 500.
Scale	double	Pixels de dispositivo por pixel lógico. Padrão 1. A saída é Round(Width * Scale) por Round(Height * Scale) pixels de dispositivo; **Chart.Scale** é definido com este valor durante a renderização e restaurado ao valor anterior depois.
CompressPNG	bool	Se a saída PNG é comprimida com deflate pelo codificador PNG do Skia. Padrão true.

Métodos

Método	Retorna	Descrição
RenderToPNGFile(string aFileName) RenderToPNGFile(string aFileName, int aWidth, int aHeight)		Renderiza para um arquivo PNG, em **Width**/**Height** ou no tamanho lógico informado, sem alterar **Width** e **Height**.
RenderToPNGStream(Stream aStream) RenderToPNGStream(Stream aStream, int aWidth, int aHeight)		Renderiza PNG (RGBA, 8 bits) para um stream.
RenderToSVG() RenderToSVG(int aWidth, int aHeight)	string	Renderiza e retorna um documento SVG 1.1 (ASCII puro) como uma string.
RenderToSVGFile(string aFileName) RenderToSVGFile(string aFileName, int aWidth, int aHeight)		Renderiza SVG para um arquivo.
RenderToSVGStream(Stream aStream) RenderToSVGStream(Stream aStream, int aWidth, int aHeight)		Renderiza SVG para um stream.

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 em pixels 0xAARRGGBB diretos (não pré-multiplicados), de cima para baixo, com stride de linha igual a **aPixelWidth**; **aPixelWidth** e **aPixelHeight** retornam com o tamanho do dispositivo.

Concorrência

Todo método de renderização pode ser chamado de qualquer thread. Cada **TsgcChartRenderer** possui suas próprias telas, uma rasterizada e uma SVG, protegidas por um bloqueio privado daquela instância do renderizador. Além disso, toda renderização realizada por qualquer **TsgcChartRenderer** no processo passa por mais um bloqueio compartilhado por todo o processo, porque **TsgcChart.Render** mantém o estado de layout no próprio gráfico enquanto executa. Enquanto uma renderização está em andamento, o gráfico não pode ser alterado estruturalmente, ou seja, nenhuma série ou eixo pode ser adicionado, removido ou descartado, e nenhuma propriedade que afete o layout pode ser alterada; adicionar dados por meio dos buffers das séries é seguro, porque as séries desenham a partir de um snapshot obtido sob o bloqueio próprio do buffer. Uma visão visual (**TsgcAvaloniaChart**, uma visão WinForms ou WPF) não adquire o bloqueio de renderização compartilhado, portanto não renderize o mesmo gráfico com um renderizador em uma thread enquanto uma visão visual o estiver desenhando no mesmo momento.

Exemplo: renderização sem interface visual para PNG no Linux

Uma aplicação de console sem referência a nenhum framework de UI, compilável e executável no Linux, que renderiza um gráfico para um arquivo PNG:
Isso é executado sem alterações em um contêiner Linux ou em um job de CI: **esegece.sgcCharts.Rendering** carrega seus próprios recursos nativos do SkiaSharp para Linux, portanto nenhum servidor X, GPU ou instalação de fontconfig é necessária.

Exemplo: pie e treemap renderizados sem interface gráfica

As séries pie, radar, heatmap, treemap e funnel (veja Series Gallery) são renderizadas sem interface gráfica como qualquer outra série. Um renderizador não desenha sobreposição ao passar o mouse, portanto a saída mostra apenas o estado mantido pelo modelo: defina-o no código, como um clique em uma visão o deixaria, com **Exploded** em uma fatia de pie ou **DrillPath** em um treemap. Um renderizador pinta vários gráficos um após o outro quando **Chart** muda:
O SVG dessas séries é idêntico, byte a byte, ao SVG que o **TsgcChartRenderer** do Delphi grava para o mesmo gráfico e tamanho. Um heatmap com **ImagePath** igual a true preenche suas células como uma única imagem, que o SVG incorpora como PNG. A demo **SeriesHeadless** em `demos\Charts` renderiza dessa forma todos os gráficos da galeria de séries: `SeriesHeadless [folder] [-dark] [-width N] [-height N] [-scale N]`.

Veja Também

- Visão Geral do .NET
- Início Rápido: WinForms
- Series Gallery
- TsgcChartRenderer (Delphi)

Series Gallery

Pacote: esegece.sgcCharts (destinado a netstandard2.0, net472, net6.0, net8.0 e net9.0)
As séries pie, radar, heatmap, treemap e funnel são os equivalentes .NET das séries Delphi de mesmo nome. Ficam no pacote principal **esegece.sgcCharts**, portanto são desenhadas em todas as visões (**TsgcWinFormsChart**, **TsgcWpfChart** e **TsgcAvaloniaChart**) e sem interface gráfica por meio do **TsgcChartRenderer** (veja Headless Rendering), com as mesmas propriedades, valores padrão e layout do Delphi. O tópico Delphi de cada série, com link abaixo, descreve cada propriedade. Esta página mostra como usá-las a partir de C# e lista os pontos em que a forma .NET da API difere.

Séries

Classe	Modo de plotagem	Descrição
TsgcChartSeries_Pie	cpmCell	Um gráfico de pizza, ou um donut quando **InnerRadiusPercent** é maior que 0. As fatias abaixo de **OthersThreshold** por cento são reunidas em uma fatia Others, **ExplodeOnClick** destaca uma fatia com um clique, e os rótulos ficam dentro ou fora com linhas de chamada. Itens: **Slices**, um **TsgcChartPieSlice** cada.
TsgcChartSeries_Radar	cpmPolar	Um polígono por série sobre raios compartilhados por todas as séries radar do gráfico, **Filled** ou apenas contorno, em uma grade poligonal ou circular (**GridShape**). Itens: **Points**, um por raio.
TsgcChartSeries_Heatmap	cpmXY	Uma grade de células coloridas por uma escala de cores: um espectrograma ao vivo alimentado linha a linha com **AppendLine** (**Capacity** mantém as linhas mais recentes), ou uma matriz fixa com linhas e colunas rotuladas em eixos de categoria. Dados: **Data**, um **TsgcChartHeatmapData**.
TsgcChartSeries_Treemap	cpmCell	Retângulos aninhados construídos a partir de caminhos como `Europe/Spain/Madrid` (**PathDelimiter**), com um cabeçalho por grupo (**Levels**), detalhamento ao clicar (**DrillOnClick**) e uma trilha de navegação para voltar. Itens: **Nodes**, um por folha.
TsgcChartSeries_Funnel	cpmCell	Uma etapa por item na ordem da coleção, afunilando até um gargalo (**NeckWidthPercent**, **NeckHeightPercent**). As etapas têm alturas iguais, ou alturas proporcionais ao valor com **ProportionalHeight**. Itens: **Stages**.

O modo de plotagem decide quais séries podem compartilhar um gráfico: veja Plot Areas. Um gráfico pode conter um pie e um funnel lado a lado, cada um em sua própria área de células, enquanto um heatmap compartilha o plot XY e seus eixos com as outras séries XY.

Itens e valores ao vivo

Membro	Descrição
AddItem(string aText, double aValue, uint aColor = 0)	Pie, radar, treemap e funnel. Acrescenta um item e retorna seu índice. Uma cor 0 usa a cor da paleta do índice do item, de modo que um item mantém sua cor enquanto os valores mudam.

SetValueAt(int aIndex, double aValue) DeleteItem(int aIndex)	Altera ou remove um item. A ordem dos itens não muda. Um índice fora do intervalo lança `ArgumentOutOfRangeException`.
ItemCount()	O número de itens, um método como toda função Delphi sem parâmetros do port.
Slices, Points, Nodes, Stages	A coleção de itens. Seu indexador retorna um **TsgcChartCategoryItem** (**Text**, **Value**, **Color**); faça o cast para **TsgcChartPieSlice** para alcançar **Exploded**.
BeginUpdate() EndUpdate()	Agrupa muitas alterações em uma única repintura e um único layout.
AppendLine(double[] aValues) AppendLine(double aKey, double[] aValues)	Heatmap. Acrescenta uma linha de células, com a chave **aKey** ou com o contador de linhas corrente do armazenamento. Os valores são um array, não uma lista `params`.
SetLineLabels(string[] aLabels) SetCellLabels(string[] aLabels)	Heatmap. Os rótulos das linhas e das células em um eixo de categoria. Uma sobrecarga recebe um **TsgcChartStrings**, a forma .NET do `TStrings` do Delphi, portanto um `null` puro é ambíguo: faça o cast.
Cells(int aLine, int aCell) SetCells(int aLine, int aCell, double aValue)	Heatmap. Lê e grava uma célula, a forma de método da propriedade indexada do Delphi `Cells[Line, Cell]`.

Interação

As visões repassam o mouse para **TsgcChart.Interaction**, portanto dicas ao passar o mouse, explode e drill funcionam da mesma forma no WinForms, WPF e Avalonia. Um clique em uma série chama primeiro **TsgcChart.OnSeriesClick** e depois a série o trata.

Membro	Descrição
TsgcChartSeries_Pie.ExplodeOnClick	Padrão false. Com true, um clique em uma fatia alterna seu sinalizador **Exploded**, ou **OthersExploded** para a fatia Others.
TsgcChartSeries_Treemap.DrillOnClick	Padrão true. Um clique em um bloco desce um nível, para o grupo sob o clique, e um clique em um item da trilha volta até ele. **DrillDown(string aPath)**, **DrillUp()**, **DrillReset()** e **DrillPath** fazem o mesmo a partir do código.
TsgcChartSeries_Treemap.OnDrill	Um **TsgcChartNotifyEvent** (`object Sender`) chamado quando **DrillPath** muda. Uma propriedade delegate, portanto atribuir um manipulador substitui o anterior, como no Delphi.
TsgcChart.OnSeriesClick	Um **TsgcChartHitEvent** (`object Sender, TsgcChartHitInfo aHit`). Para essas séries, **aHit.SubIndex** é o índice do item (-2 para a fatia Others de um pie), **aHit.Text** o texto da fatia ou da etapa ou o caminho do treemap, e **aHit.Percent** a participação do item em um pie, um funnel ou um treemap.

A sobreposição ao passar o mouse (o destaque e a dica) só é desenhada na tela. **TsgcChartRenderer**, **SaveToPNG** e **SaveToSVG** renderizam apenas o modelo, de modo que uma fatia destacada ou um treemap detalhado aparece na saída, enquanto uma fatia sob o mouse não aparece.

Exemplo: construindo as cinco séries

Uma classe auxiliar que constrói um gráfico por série, os mesmos gráficos da demo **SeriesGallery** com menos dados:

Exemplo: hospedando os gráficos em um formulário WinForms

Um **TsgcWinFormsChart** por aba, uma legenda que acompanha os cliques e o caminho do drill, e valores ao vivo. Com WPF use **TsgcWpfChart**, com Avalonia **TsgcAvaloniaChart**: o código do gráfico não muda.

Demos

Demo	Framework	Descrição
SeriesGallery	WinForms, net8.0-windows	O pie, o radar, um espectrograma de 30 linhas por segundo, um heatmap de matriz, o treemap e o funnel, um **TsgcWinFormsChart** por aba, com um botão de donut, opções de radar preenchido ou circular e botões Up e Top para o treemap.
SeriesGalleryAvalonia	Avalonia, net8.0	A mesma galeria em controles **TsgcAvaloniaChart**, no Windows, Linux e macOS.
SeriesHeadless	Console, net8.0	Os gráficos da galeria renderizados com **TsgcChartRenderer** em arquivos PNG e SVG, sem janela: `SeriesHeadless [folder] [-dark] [-width N] [-height N] [-scale N]` .

As demos estão em `demos\Charts`. Ambas as galerias aceitam `-seconds N` como autoteste: mostram cada aba, clicam em uma fatia do pie e em um bloco do treemap pela interação do gráfico, fecham após N segundos e retornam o código de saída 0 quando todas as verificações passam.

Veja Também

- .NET Overview
- .NET Differences from Delphi
- Headless Rendering
- Quick Start: WinForms
- Plot Areas
- Colour Scale

Bridge

O pacote **esegece.sgcCharts.sgcWebSockets** (destinado a `netstandard2.0`, `net8.0`, `net9.0`) conecta as fontes de dados do sgcCharts a uma conexão sgcWebSockets ativa: mensagens WebSocket, publicações MQTT, Server-Sent Events e streams de mercado da Binance. É um pacote separado do pacote principal **esegece.sgcCharts** e exige o sgcWebSockets para compilar; um projeto que usa apenas as fontes de dados principais não precisa deste pacote nem do sgcWebSockets.
Por padrão, o pacote referencia diretamente o projeto da porta .NET do sgcWebSockets; uma compilação sem essa árvore de código-fonte pode, em vez disso, referenciar o pacote NuGet **esegece.sgcWebSockets**. Esta ponte não é distribuída na edição Community.

MQTT: TsgcChartDataSource_MQTT

Exibe publicações MQTT recebidas por um **TsgcWSProtocol_MQTT_Client**. Enquanto a fonte está **Active** e **Client** está atribuído, ela inscreve seu próprio handler em **Client.OnMQTTPublish**.

Propriedade	Tipo	Descrição
Client	TsgcWSProtocol_MQTT_Client	O cliente MQTT cujas publicações são exibidas no gráfico.
TopicFilter	string	Filtro de tópico MQTT, com os curingas `+` (um nível) e `#` (este e todo nível inferior). Um filtro vazio exibe toda publicação.
PayloadFormat	TsgcChartMQTTPayloadFormat (cpfJSON, cpfNumber)	**cpfJSON** lê o payload por meio de **Mapping** (um objeto JSON ou um array de registros). **cpfNumber** interpreta o payload como um número simples para Y e usa a hora local atual para X.
Active	bool (oculta a propriedade base)	A fonte redeclara **Active** para que defini-la também atualize o handler em **Client**, da mesma forma que atribuir **Client** faz.

WebSocket: TsgcChartDataSource_WebSocket

Exibe mensagens JSON recebidas por um **TsgcWebSocketClient**. Enquanto a fonte está **Active** e **Client** está atribuído, ela inscreve seu próprio handler em **Client.OnMessage**. Toda mensagem de texto é lida como JSON por meio de **Mapping**.

Propriedade	Tipo	Descrição
Client	TsgcWebSocketClient	O cliente WebSocket cujas mensagens são exibidas no gráfico.
MessageFilter	string	Um caminho JSON que deve existir em uma mensagem para que ela seja exibida, por exemplo `"k"` para manter apenas mensagens de kline da Binance. Um filtro vazio exibe toda mensagem que seja interpretada como JSON.
Active	bool (oculta a propriedade base)	Redeclarada como em **TsgcChartDataSource_MQTT**, de modo que o vínculo segue o valor.

Server-Sent Events: TsgcChartDataSource_SSE

Exibe Server-Sent Events de duas fontes independentes, que podem ser usadas juntas na mesma fonte de dados.

- **Client** é um **TsgcHTTP1Client**, como no Delphi. A aplicação inicia o stream com o método bloqueante do cliente `GetSSE(URL)`, normalmente a partir de uma thread de trabalho. Enquanto a fonte está **Active** e **Client** está atribuído, ela inscreve seu próprio handler em **Client.OnSSEMessage**, que entrega o texto bruto do stream conforme ele chega. A fonte armazena os blocos em buffer, os divide em eventos a cada linha em branco, junta as linhas `data:` de um evento, e lê o resultado como JSON por meio de **Mapping**.
- **SSEClient** é um **TsgcSSEClient**, o cliente gerenciado no estilo EventSource. Não tem contraparte em Delphi nesta ponte. Seu **OnEvent** entrega eventos já interpretados, portanto a fonte lê o **Data** do evento como JSON por meio de **Mapping** diretamente, sem envolver a interpretação de blocos.

Propriedade	Tipo	Descrição
Client	TsgcHTTP1Client	Cliente SSE bloqueante. Veja acima.
SSEClient	TsgcSSEClient	Cliente EventSource gerenciado, um adicional do .NET. Veja acima.
Active	bool (oculta a propriedade base)	Redeclarada como em **TsgcChartDataSource_MQTT**. Tanto **Client** quanto **SSEClient** são vinculados ou desvinculados independentemente conforme a propriedade muda.

Um método **ResetStream** descarta um evento em buffer, ainda incompleto, do interpretador de blocos de **Client**, por exemplo após uma reconexão.

Binance: TsgcChartDataSource_Binance

Assina e exibe um stream público de dados de mercado da Binance por meio de um **TsgcWSAPI_Binance**. Os streams de mercado da Binance não têm evento tipado: eles chegam como JSON bruto no evento **OnMessage** do cliente WebSocket ao qual a API está conectada (**API.Client**). Enquanto a fonte está **Active** e **API** está atribuído, ela inscreve seu próprio handler ali; o vínculo segue **API.Client** sempre que **API** ou **Active** é definido, e também em **Subscribe**.

Propriedade	Tipo	Descrição
API	TsgcWSAPI_Binance	O componente da API Binance cujo stream é exibido no gráfico.
Symbol	string	O símbolo Binance, por exemplo `"BTCUSDT"`. Convertido para minúsculas automaticamente para a assinatura, já que os nomes de stream da Binance são apenas em minúsculas. Mensagens para outro símbolo são ignoradas.
Stream	TsgcChartBinanceStream (cbsKLine, cbsTicker, cbsTrade)	Qual evento Binance é exibido: klines (candles), o ticker de 24 horas (último preço), ou negociações individuais.
Interval	TsgcWSBinanceChartIntervals	Intervalo do kline, por exemplo `bci1m`. Usado apenas para **cbsKLine**; mensagens para outro intervalo são ignoradas.
ClosedOnly	bool	Somente **cbsKLine**. Quando true, apenas candles fechados são exibidos; o candle ainda em formação é, do contrário, atualizado no lugar a cada mensagem.
Aggregator	TsgcChartAggregator (edição financeira)	Somente **cbsTrade**. Quando atribuído, toda negociação vai para **Aggregator.AddTrade** em vez do buffer da série: a negociação em si não é plotada, e a série OHLC do agregador desenha os candles (volume incluído). Descartar o agregador não limpa esta proprieda-

		de; um agregador descartado ignora as negociações que ainda recebe.
Active	bool (oculta a propriedade base)	Redeclarada como em **TsgcChartDataSource_MQTT**.

Eventos

Evento	Tipo	Descrição
OnTrade	TsgcChartBinanceTradeEvent(object Sender, double aTime, double aPrice, double aQty)	Somente **cbsTrade**. Dispara para toda negociação de **Symbol**, com sua hora UTC, preço e quantidade, antes que a negociação seja adicionada ao buffer ou repassada ao **Aggregator**.

Métodos

Método	Retorna	Descrição
Subscribe()	int	Assina **Stream** para **Symbol** (e **Interval** para klines) chamando `SubscribeKLine`, `SubscribeTicker` ou `SubscribeTrades` de **API**. Retorna o id que a API retorna.
Unsubscribe()	int	A chamada de cancelamento de assinatura correspondente.

Eventos multicast

Toda fonte de dados da ponte vincula um evento do cliente que é um simples ponteiro de método na biblioteca Delphi (**OnMQTTPublish**, **OnMessage**, **OnSSEMessage**, **OnEvent**), mas aqui um evento multicast do .NET. A fonte Delphi salva o handler já atribuído, instala o próprio, e restaura o salvo ao se desvincular. Uma fonte de dados .NET, em vez disso, adiciona seu próprio handler à lista de invocação com `+=` ao se vincular, e remove apenas esse mesmo handler com `-=` ao se desvincular; todo handler da aplicação já inscrito, ou inscrito posteriormente, continua disparando e nunca é tocado pela fonte.
A diferença prática é a ordem de inscrição, não a presença: os handlers executam na ordem em que foram adicionados, portanto um handler da aplicação inscrito no cliente antes que a fonte de dados seja vinculada executa antes que a fonte de dados exiba a mensagem, ao passo que o Delphi sempre exibe primeiro. A vinculação ocorre apenas em tempo de execução, nunca em tempo de design. Não há **FreeNotification** no .NET, portanto descartar o cliente subjacente não limpa a propriedade **Client** (ou **API**, ou **SSEClient**) da fonte de dados; defina-a como null, ou descarte a fonte de dados, em vez disso. Em **TsgcChartDataSource_SSE**, um handler da aplicação em **OnSSEMessage** ainda pode definir **Cancel**, que é compartilhado por toda a lista de invocação, de modo que qualquer handler que o defina como true interrompe o stream para todos os demais handlers também.
Mensagens, publicações e eventos chegam sempre na thread a partir da qual o cliente subjacente notifica. As adições passam pelo bloqueio e pela notificação limitada da **TsgcChartDataSource** base, portanto qualquer modo de notificação é seguro de usar com uma fonte de dados de ponte.

Exemplo: gráfico ao vivo da Binance

Veja Também

- Visão Geral da Bridge (Delphi)
- Visão Geral do .NET

Diferenças do .NET em Relação ao Delphi

O sgcCharts .NET mantém os nomes de classes, propriedades e comportamento do Delphi sempre que as duas linguagens permitem (veja Visão Geral do .NET). Esta página lista os pontos em que a porta .NET realmente se comporta de forma diferente da biblioteca Delphi, não apenas onde um nome ou um tipo mudou para seu equivalente natural em C#.

Eventos

Área	Delphi	.NET
Fontes de dados de ponte ao vivo (**esegece.sgcCharts.sgcWebSockets**): **OnMQTTPublish**, **OnMessage** (cliente WebSocket e cliente Binance), **OnSSEMessage** / **OnEvent**	Um único ponteiro de método. A fonte de dados salva o handler atual do cliente, instala o próprio, e restaura o salvo quando para.	Um `event` multicast real do C# (`+=` / `-=`). A fonte de dados inscreve e cancela a inscrição do próprio handler em vez de salvar e restaurar um; veja Visão Geral da Bridge para o mecanismo completo.
Modelo de gráfico principal: **TsgcChart.OnChange**, **TsgcChartDataSource.OnDataChanged** / **OnBeforeAppend**, **TsgcChart.OnCrosshairMove** / **OnSeriesClick** / **OnZoom**, e membros semelhantes em eixos, séries e desenhos	Um único ponteiro de método, substituído por atribuição.	Sem diferença: estes permanecem como uma simples propriedade delegada `get; set;` , não um `event` do C#, portanto atribuir um novo handler ainda substitui o anterior exatamente como no Delphi. Listado aqui apenas para contrastar com a linha da bridge acima, já que ambos se chamam **OnX**.

Propriedades indexadas

Área	Delphi	.NET
TsgcChartAxes, **TsgcChartPanes**, **TsgcChartDrawings**, **TsgcChartPalette**, **TsgcChartJSONNode**	Uma propriedade de array indexada, por exemplo `Axes[Index]` .	Um indexador real do C#: `this[int Index]` .
TsgcChartBuffer (`X` , `Y`), **TsgcChartOHLCBuffer** (`X` , `Open` , `High` , `Low` , `Close` , `Volume`), os `XValues` e `Values` das séries financeiras	Uma propriedade de array indexada por coluna, por exemplo `Open[Index]` , e uma bidimensional, `Values[Line, Index]` .	Um método comum que recebe o índice como parâmetro: `Open(Index)` , `Values(Line, Index)` . Uma classe C# só pode declarar uma assinatura de indexador (`this[...]`), e essas classes precisam de vários acessores por índice, nomeados independentemente ou com múltiplos parâmetros, na mesma classe, portanto se tornam métodos em vez disso.

As séries pie, radar, heatmap, treemap e funnel e suas classes auxiliares: **TsgcChartSeries_Radar.SpokeText**, **TsgcChartSeries_Funnel.DrawnStages**, **TsgcChartSeries_Heatmap.Cells**, **TsgcChartHeatmapData.Cells**, **TsgcChartColorScale.LUT**, **TsgcChartPieLayout.Slices**, **TsgcChartTreemapLayout.Nodes**, **Crumbs** e **DrawnNodes**	Uma propriedade array indexada, por exemplo `SpokeText[Index]`, `DrawnStages[Index]`, e a de leitura e escrita `Cells[Line, Cell]`.	Um método: `SpokeText(Index)`, `DrawnStages(Index)` (uma cópia do registro da etapa), `Cells(Line, Cell)` para ler e `SetCells(Line, Cell, Value)` para gravar. `DrawnStages` lança `ArgumentOutOfRangeException` onde o Delphi levanta `EListError`. Os métodos de layout `Slices`, `Nodes`, `Crumbs` e `DrawnNodes` também a lançam para um índice fora de `0..Count - 1`, onde o Delphi lê o array interno sem verificação.

Séries pie, radar, heatmap, treemap e funnel

Área	Delphi	.NET
SetValueAt e **DeleteItem** das séries pie, radar, treemap e funnel	Um índice fora do intervalo levanta `ERangeError`.	Um índice fora do intervalo lança `ArgumentOutOfRangeException`.
TsgcChartSeries_Heatmap: **AppendLine**, **SetLineLabels**, **SetCellLabels**	Um parâmetro array aberto (`AppendLine([1, 2, 3])`) e uma sobrecarga `TStrings` para os rótulos.	Um parâmetro `double[]` ou `string[]`, não uma lista `params`, e uma sobrecarga **TsgcChartStrings** para os rótulos. Um `null` puro é ambíguo entre as duas sobrecargas de rótulos, portanto faça o cast.
Os itens (**Slices**, **Points**, **Nodes**, **Stages**) e os **Levels** do treemap	Editados no Object Inspector e armazenados no formulário (DFM).	Não há streaming do designer de formulários: adicione os itens no código com **AddItem** e os níveis com **Levels.Add()**. Veja Series Gallery.
Saída raster (**RenderToPNGFile**, **RenderToBuffer**, **SaveToPNG**)	Desenhada pelo canvas raster do sgcUICore.	Desenhada pelo SkiaSharp, portanto os pixels suavizados da borda podem diferir ligeiramente do Delphi. O layout é o mesmo, e a saída SVG é igual ao SVG do Delphi byte a byte.

Carregando de um DataSet ou de um DataReader

TsgcChartDataSource_DataSet carrega a partir de um `System.Data.DataTable` ou de qualquer `IDataReader`, os equivalentes .NET do `TDataSet` do Delphi. Um `DataTable` não tem cursor, portanto não há registro corrente para salvar e restaurar após o carregamento: o tratamento de bookmark da fonte de dados Delphi, que retorna o `TDataSet` de origem ao registro em que estava antes de **Refresh** executar, não tem contraparte no .NET.

Edições

O sgcCharts Delphi distribui as edições Community, Trial e Registered. O sgcCharts .NET distribui apenas **Community** (compilado com `-p:SgcCommunity=true`) e **Registered** (a compilação padrão): não há compilação trial nem constante trial no .NET.

Veja Também

- Visão Geral do .NET
- Series Gallery
- Visão Geral da Bridge
- TsgcChartDataSource_DataSet

Solução de Problemas

Nenhum problema conhecido está documentado ainda.

Introdução

sgcMaps é uma biblioteca de componentes Delphi para mapas interativos. Os controles de mapa desenham tiles raster e vetoriais de forma nativa, sem controle de navegador e sem JavaScript, e exibem marcadores, rotas, formas, rótulos e mapas de calor sobre eles. Componentes não visuais adicionam geocodificação, direções, busca de lugares, elevação, fusos horários e localização do dispositivo, e o roteador offline e o geocodificador offline respondem sem nenhuma conexão de rede.
Um controle VCL, um controle FireMonkey e um controle Lazarus (LCL) compartilham um único modelo de mapa, de modo que o código que adiciona uma camada, um marcador ou uma rota é o mesmo em todos os frameworks. O controle Lazarus não consegue baixar tiles, portanto no Lazarus ele exibe apenas tiles de uma pasta. Um controle de mapa também pode renderizar sem janela, o que transforma uma aplicação de console ou um serviço em um renderizador de mapas estáticos.

- Visão Geral: o que o sgcMaps inclui e qual versão do Delphi cada recurso exige.
- Instalação: o instalador sgcUI, os pacotes e os requisitos.
- Início Rápido VCL e Início Rápido FMX: um mapa funcionando em poucas linhas.
- Controles de mapa: TsgcVCLMap e TsgcFMXMap, suas camadas e sobreposições.
- Serviços online: geocodificação, direções, lugares, elevação e fusos horários.
- Roteamento: edição interativa de rotas e o roteador offline.
- Geocodificação offline: busca de lugares a partir de um índice GeoNames local.
- Localização: a posição do dispositivo, real ou simulada.
- .NET: sgcMaps for .NET, a porta em C# com um controle Windows Forms, renderização sem interface visual e os serviços offline.
- Demos, Ferramentas, Lazarus e Solução de Problemas: os projetos de exemplo, as ferramentas de dados, o controle LCL e as armadilhas mais comuns.

Visão Geral

O sgcMaps é construído em torno de um **modelo de mapa**: um conjunto de coleções que descrevem o que um mapa exibe. Layers guarda as fontes de tiles, e Markers, Polylines, Polygons, Rectangles, Circles, Labels e Heatmaps guardam as sobreposições. Options e MarkerOptions definem o comportamento da visualização. As classes do modelo ficam em **sgcMaps_Classes** e são as mesmas em TsgcVCLMap e TsgcFMXMap.
Os componentes não visuais trabalham com essas coleções, e não com um controle específico. TsgcMapRouteCalculator mantém seus pontos de passagem nos Markers de um mapa e sua rota nas Polylines, e TsgcMapLocation mantém atualizados um marcador de posição e um círculo de precisão. Por isso, o mesmo componente atende a um formulário VCL e a um formulário FireMonkey.

Recursos

Tiles e camadas

- Tiles raster online de provedores integrados: OpenStreetMap, OpenTopoMap, Esri World Imagery, Esri World Street, CARTO Light e Dark, MapTiler Streets e Satellite, Thunderforest Cycle e Transport, e Stadia Alidade Smooth e Smooth Dark.
- Qualquer outro servidor de tiles por meio de um modelo de URL com marcadores de posição {z}, {x}, {y}, subdomínio, retina e chave de API.
- Um cache em disco por camada, de modo que os tiles já vistos sejam desenhados sem conexão.
- Tiles offline de uma pasta z/x/y ou de um arquivo MBTiles.
- Tiles vetoriais (Mapbox Vector Tiles) de um servidor ou de um arquivo MBTiles local, desenhados com um estilo MapLibre ou Mapbox GL.
- Várias camadas com opacidade, faixa de zoom, papel de base ou de sobreposição, e um seletor de camadas opcional no mapa.
- Um baixador de tiles que preenche uma pasta com todos os tiles de uma área e faixa de zoom, para um pacote de mapa offline.

Sobreposições

- Marcadores com texto de popup e de dica de ferramenta, cor, bitmaps de ícone personalizados, âncoras e arrastar.
- Agrupamento de marcadores, para que milhares de marcadores continuem legíveis e rápidos.
- Polilinhas (contínuas, tracejadas ou geodésicas), polígonos com furos, retângulos e círculos, todos editáveis com o mouse.
- Rótulos de texto e mapas de calor ponderados com vários gradientes de cor.

Interação e saída

- Zoom por arrastar, pela roda do mouse e por botões, rotação do mapa (bearing), uma barra de escala e a atribuição dos tiles.
- Eventos para cliques, arrastes de marcadores, edições de formas, cliques em clusters, mudanças de camada, mudanças de zoom e erros de tiles.
- Renderização para um bitmap, um arquivo PNG ou BMP, também a partir de um controle que não tem janela pai.

Dados

- Importação e exportação de GeoJSON, GPX e KML de marcadores, polilinhas e polígonos.
- Funções de geometria: distância, bearing, ponto de destino, interpolação geodésica, ponto em polígono, simplificação de linhas e ponto mais próximo em uma polilinha.
- Um leitor em streaming de arquivos .osm.pbf e XML .osm do OpenStreetMap.

Serviços

- Geocodificação e geocodificação reversa com Nominatim, Google, HERE, Azure Maps, Mapbox, OpenRouteService e Geoapify.
- Direções com OSRM, OpenRouteService, Google Routes, HERE, Azure Maps, Mapbox e Geoapify.
- Preenchimento automático de lugares com Photon, Nominatim, Google, HERE, Mapbox e Geoapify.
- Elevação de pontos e caminhos com Open-Elevation, OpenTopoData e Google.
- Fuso horário de uma coordenada com TimeAPI, GeoNames, Google e Azure Maps.

Roteamento, dados offline e localização

- Edição interativa de rotas com pontos de passagem arrastáveis e rotas alternativas.
- Roteamento offline para carro, bicicleta e a pé em um grafo construído a partir de dados do OpenStreetMap.
- Geocodificação offline em um índice SQLite construído a partir dos dumps do GeoNames.
- Localização do dispositivo no Windows, Android, iOS e macOS, além de simulação de posição e de trajeto para testes.

Versões do Delphi

O modelo de mapa, o controle VCL, GPX, KML e as funções de geometria compilam do Delphi 7 ao Delphi 13, e os pacotes cobrem também o C++Builder 2007 ao C++Builder 13. Alguns recursos dependem de partes da RTL que chegaram depois, por isso seus componentes são registrados apenas nas versões do IDE que as possuem.

Recurso	Unit	Versão mínima
Controle de mapa VCL, sobreposições, GPX, KML, geometria	sgcMaps_VCL, sgcMaps_Classes	Delphi 7
Saída PNG	sgcMaps_VCL	Delphi 2009
Tiles de um servidor, baixador de tiles	sgcMaps_TileSource_HTTP, sgcMaps_TileDownloader	Delphi 2009 (Indy 10; THTTPClient a partir do Delphi XE8)
Calculador de rotas, roteador offline, tiles vetoriais de um servidor	sgcMaps_RouteCalculator, sgcMaps_OfflineRouter, sgcMaps_TileSource_MVT	Delphi XE2
Controle de mapa FireMonkey, localização do dispositivo, GeoJSON	sgcMaps_FMX, sgcMaps_Location, sgcMaps_GeoJSON	Delphi XE6
MBTiles, MBTiles vetorial, geocodificador offline	sgcMaps_TileSource_MBTiles, sgcMaps_OfflineGeocoder	Delphi XE7 (FireDAC SQLite)
Serviços online	sgcMaps_Geocoding, sgcMaps_Directions, sgcMaps_Places, sgcMaps_Elevation, sgcMaps_TimeZone	Delphi XE8 (THTTPClient, System.JSON)

No Lazarus o pacote contém o controle de mapa LCL, a fonte de tiles de pasta, os provedores, o motor raster e as funções de geometria. O controle LCL não consegue baixar tiles, portanto exibe apenas tiles de uma pasta, e GeoJSON, GPX e KML não são compilados no FPC. Veja Lazarus.

.NET

O sgcMaps também tem uma porta .NET, o sgcMaps for .NET, para o .NET 8 e o .NET 9. Ela tem o mesmo modelo de mapa, os mesmos serviços online e os mesmos arquivos de dados offline, um controle Windows Forms e um renderizador sem interface visual que também roda no Linux e no macOS. Veja a visão geral do .NET.

Instalação

O sgcMaps vem no **instalador sgcUI** unificado, junto com os outros produtos visuais da eSeGeCe. O instalador sabe quais produtos sua licença inclui, portanto não há nada a escolher manualmente além dos IDEs.

Instalando com o instalador sgcUI

1. Feche todas as cópias abertas do IDE e inicie o instalador sgcUI.
2. Faça login com sua conta eSeGeCe.
3. O instalador lê sua licença e lista os produtos que ela inclui. O sgcMaps vem marcado quando sua licença o cobre.
4. Selecione as versões do Delphi e do C++Builder em que instalar e inicie a instalação.
5. Abra o IDE. Os componentes estão na página **SGC Maps** da Paleta de Ferramentas.

Os controles de mapa aparecem apenas na paleta do seu próprio framework: TsgcVCLMap em projetos VCL e TsgcFMXMap em projetos FireMonkey. Os componentes não visuais aparecem em ambas.

Requisitos

- A partir do Delphi XE7, o pacote de runtime exige o **FireDAC** com seu driver **SQLite** (os pacotes FireDAC, FireDACCommon, FireDACCommonDriver e FireDACSqliteDriver). O sgcMaps o utiliza para ler arquivos MBTiles e o índice do geocodificador offline, portanto sua edição do Delphi deve incluir o FireDAC SQLite.
- Os serviços online usam THTTPClient (System.Net.HttpClient) e System.JSON, disponíveis a partir do Delphi XE8.
- TsgcMapLocation usa System.Sensors, disponível a partir do Delphi XE6.

Pacotes

A pasta **Packages** contém um par de pacotes por versão do IDE: o pacote de runtime **sgcMapsD*nn*.dpk** e o pacote de design **dclsgcMapsD*nn*.dpk**, do Delphi 7 (D7) ao Delphi 13 (D13). O C++Builder usa os projetos correspondentes **sgcMapsC*nn*.cbproj** e **dclsgcMapsC*nn*.cbproj**, do C++Builder 2007 ao C++Builder 13. Para instalar manualmente, compile primeiro o pacote de runtime, depois instale o pacote de design e adicione a pasta Source ao library path do IDE.

Lazarus

O Lazarus tem seu próprio pacote, **Packages\Lazarus\sgcMapsLazarus.lpk**. Abra-o com Package, Open Package File, compile-o e escolha Use, Install, o que reconstrói o IDE. Ele exige os pacotes LCL, FCL e IDEIntf e contém o controle de mapa (TsgcVCLMap na LCL), a fonte de tiles de pasta, os provedores, o motor raster e as funções de geometria. O controle LCL não consegue baixar tiles, portanto exibe apenas tiles de uma pasta. GeoJSON, GPX e KML constam do pacote, mas não são compilados no FPC, e os serviços online, o calculador de rotas, o roteador offline, o geocodificador offline e o TsgcMapLocation estão disponíveis somente no Delphi. O tópico Lazarus traz os detalhes e um exemplo.

Units

Adicione à cláusula uses a unit de cada componente que você cria em código. Componentes colocados em um formulário adicionam suas units automaticamente.

Unit	Conteúdo
sgcMaps_Classes	O modelo de mapa: camadas, marcadores, polilinhas, polígonos, retângulos, círculos, rótulos, mapas de calor e opções, além de TsgcMapCoord.

sgcMaps_VCL	TsgcVCLMap.
sgcMaps_FMX	TsgcFMXMap.
sgcMaps_Providers	Os provedores de tiles integrados e sgcMapApplyProvider.
sgcMaps_Services	TsgcMapService, a classe base dos serviços online.
sgcMaps_Geocoding	TsgcMapGeocoder e as classes de resultado de geocodificação.
sgcMaps_Directions	TsgcMapDirections, as classes de resultado de rota e sgcMapAddRouteToPolylines.
sgcMaps_Places	TsgcMapPlaces.
sgcMaps_Elevation	TsgcMapElevation.
sgcMaps_TimeZone	TsgcMapTimeZone.
sgcMaps_RouteCalculator	TsgcMapRouteCalculator.
sgcMaps_OfflineRouter	TsgcMapOfflineRouter e TsgcMapRouteGraphBuilder.
sgcMaps_OfflineGeocoder	TsgcMapOfflineGeocoder e TsgcMapGeoIndexBuilder.
sgcMaps_Location	TsgcMapLocation.

.NET

O instalador do sgcUI não instala nada para .NET. O sgcMaps for .NET é distribuído como seus próprios pacotes NuGet (esegece.sgcMaps e seus pacotes complementares), somente para a edição Registered. Veja Instalação do .NET.

Início Rápido VCL

Este passo a passo mostra um mapa em um formulário VCL, primeiro com o designer de formulários e depois inteiramente em código.

1. Em tempo de design

1. Crie uma aplicação VCL e solte um **TsgcVCLMap**, da página SGC Maps, no formulário. Defina sua propriedade **Align** como alClient.
2. Adicione a camada base no evento **OnCreate** do formulário com **Layers.Add** e defina sua propriedade **Provider**. Uma camada com provedor baixa os tiles desse provedor sem nenhuma outra configuração. A chamada opcional a **sgcMapApplyProvider** também copia para a camada o tipo de fonte, o modelo de URL, o zoom máximo e a atribuição do provedor.
3. Defina a visualização inicial com **SetCenter** e **Zoom**, e adicione sobreposições com **AddMarker**, **AddPolyline** ou **AddCircle**.
4. Execute a aplicação. Arraste o mapa para movê-lo e use a roda do mouse ou os botões de zoom para aplicar zoom.

O manipulador OnCreate completo para esses passos é o exemplo do tópico TsgcVCLMap.

2. Em código

O programa abaixo constrói o mesmo mapa sem arquivo .dfm. O formulário cria seu próprio controle, e por isso o construtor chama **CreateNew** em vez do **Create** herdado. Um clique no marcador abre seu popup, e um clique em qualquer outro lugar adiciona um novo marcador naquela posição.

```
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.
```

Sobre os tiles

A camada guarda em **CacheFolder** todo tile que baixa, de modo que uma visualização já vista seja desenhada de imediato e sem conexão. Os servidores de tiles do OpenStreetMap são mantidos por voluntários e têm uma política de uso que não permite uso intenso. Para uma aplicação em produção, use um provedor comercial como MapTiler, Thunderforest ou Stadia (defina o **ApiKey** da camada), seu próprio servidor de tiles, ou tiles offline de uma pasta ou de um arquivo MBTiles. Mantenha **Options.ShowAttribution** habilitado, pois os provedores de tiles exigem que a atribuição esteja visível.

Próximos passos

- TsgcVCLMap: todas as propriedades, métodos e eventos do controle.
- TsgcMapGeocoder: pesquise um endereço e mostre-o no mapa.
- TsgcMapRouteCalculator: permita que o usuário construa uma rota clicando no mapa.

Início Rápido FMX

TsgcFMXMap é o controle de mapa FireMonkey. Ele compartilha o modelo de mapa com o controle VCL, de modo que camadas, marcadores e formas são criados exatamente como no Início Rápido VCL. Exige o Delphi XE6 ou posterior e roda nas plataformas desktop e móveis do FireMonkey.

1. Em tempo de design

1. Crie uma aplicação multi-dispositivo e solte um **TsgcFMXMap**, da página SGC Maps, no formulário. Defina sua propriedade **Align** como Client.
2. Adicione a camada base no evento **OnCreate** do formulário com **Layers.Add** e defina sua propriedade **Provider**, o que basta para baixar os tiles desse provedor. A chamada opcional a **sgcMapApplyProvider** também copia para a camada o tipo de fonte, o modelo de URL, o zoom máximo e a atribuição do provedor. Em um dispositivo móvel a pasta da aplicação é somente leitura, portanto coloque o cache de tiles em uma pasta gravável, como **TPath.GetCachePath**.
3. Defina a visualização inicial com **SetCenter** e **Zoom**, e adicione marcadores.
4. Execute a aplicação. Mova com um dedo, aplique zoom com o gesto de pinça e gire com dois dedos quando **Options.RotationEnabled** for True.

O manipulador OnCreate completo para esses passos é o exemplo do tópico TsgcFMXMap.

2. Em código

O programa abaixo constrói o mapa em código. Ele ativa o agrupamento de marcadores, de modo que marcadores próximos se fundem em um círculo numerado quando o mapa é reduzido, e mostra o popup de um marcador quando ele é tocado.

```
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.
```

Próximos passos

- TsgcFMXMap: o que difere do controle VCL.
- TsgcMapLocation: mostre e acompanhe a posição do dispositivo.
- TsgcMapPlaces: sugira lugares enquanto o usuário digita.

Demos

A pasta **Demos\Maps** da instalação do sgcUI tem sete projetos Delphi. Cada um é uma aplicação pequena e completa que mostra um grupo de recursos, e juntos eles usam todos os componentes do sgcMaps.

Demo	Framework	Versão mínima	Mostra
01 Offline Map	VCL	Delphi 7	Tiles de pasta, MBTiles e online, marcadores, formas, importação de GeoJSON, GPX e KML.
02 Map Features	VCL	Delphi 7	Provedores e chaves de API, agrupamento, edição, mapas de calor, rotação e exportação.
03 Services	VCL	Delphi XE8	Busca de lugares, geocodificação, direções, elevação, fusos horários e localização.
04 Offline Services	VCL	Delphi XE7	Tiles offline, o geocodificador offline e o roteador offline.
05 Static Render	Console	Delphi 7	Um mapa renderizado em um arquivo PNG ou BMP sem janela.
06 FireMonkey	FireMonkey	Delphi XE6	O controle FireMonkey no desktop e no mobile, com localização, busca e roteamento.
07 Vector Tiles	VCL	Delphi XE2	Tiles vetoriais de um servidor ou de um arquivo MBTiles local, com estilos, retina e rotação.

Os demos não têm chaves de API no código-fonte. Os que usam um provedor que exige chave têm uma caixa de edição para ela, e todo demo começa com serviços e tiles que não precisam de chave.
A maioria dos demos baixa tiles dos servidores públicos do OpenStreetMap, o que serve para experimentar os demos, mas não para uma aplicação em produção. Veja Solução de Problemas para as políticas de uso.

01 Offline Map

Pasta	Projeto	Fra-mework	Versão mínima
Demos\Maps\01.OfflineMap	sgcMapsDemo.dpr	VCL	Delphi 7 (tiles online a partir do Delphi 2009, GeoJSON a partir do Delphi XE6, MBTiles a partir do Delphi XE7)

O que mostra

Um TsgcVCLMap cuja camada base pode vir de três fontes, escolhidas com as opções **Folder**, **MBTiles** e **Online**: uma pasta z/x/y de tiles, um arquivo MBTiles ou o servidor de tiles do OpenStreetMap com cache em disco. O mapa começa em Madri com zoom 13.

- **Go To Coordinates** e uma caixa de zoom movem a visualização.
- **Marker at Center**, **Sample Markers** e **Clear Markers** adicionam e removem marcadores.
- **Sample Circle**, **Sample Polyline** e **Sample Polygon** desenham formas.
- As caixas de seleção **Zoom Controls** e **Scale Bar** ligam e desligam essas partes do mapa.
- **GeoJSON...**, **GPX...** e **KML...** importam um arquivo para o mapa.
- O log mostra os eventos OnMapClick, OnMarkerClick e OnZoomChanged.

Componentes e units

TsgcVCLMap com camadas de pasta, MBTiles e HTTP, e as funções sgcMapLoadGeoJSONFile, sgcMapLoadGPXFile e sgcMapLoadKMLFile. Units: sgcMaps_Classes, sgcMaps_TileSource, sgcMaps_VCL, sgcMaps_GeoJSON, sgcMaps_GPX e sgcMaps_KML.

Como executar

Abra **sgcMapsDemo.dpr** no IDE, depois compile e execute. O demo encontra as units do sgcMaps pelo library path definido pelo instalador sgcUI; com uma instalação manual, adicione primeiro a pasta Source ao library path.

Dados e chaves

- Nenhuma chave de API é necessária.
- A pasta **data** ao lado do projeto tem sample.geojson, sample.gpx e sample.kml para os botões de importação. O demo a procura primeiro ao lado do executável e depois na pasta atual.
- Um caminho relativo de tiles é resolvido a partir da pasta do executável. O caminho padrão é **tiles**, uma pasta de arquivos z\x\y.png, que você pode preencher com o baixador de tiles (veja Tiles Offline e o Baixador de Tiles).
- Os tiles online ficam em cache na pasta **tile_cache** ao lado do executável.
- Quando a pasta de tiles ou o arquivo MBTiles não existe, o demo escreve uma linha no log.
- Em versões do Delphi anteriores ao XE6 o botão GeoJSON fica desabilitado, e em versões anteriores ao XE7 a opção MBTiles apenas registra no log que precisa do FireDAC.

02 Map Features

Pasta	Projeto	Framework	Versão mínima
Demos\Maps\02.MapFeatures	sgcMapsFeaturesDemo.dpr	VCL	Delphi 7 (tiles online e PNG a partir do Delphi 2009, GeoJSON a partir do Delphi XE6)

O que mostra

Os recursos de desenho e de interação do TsgcVCLMap, uma aba para cada:

Aba	O que você pode fazer
Layers	Escolha uma camada base, digite uma chave de API do MapTiler, Thunderforest ou Stadia e pressione **Apply key** para adicionar suas camadas, e mostre o seletor de camadas no mapa.
Clusters	Adicione 2000 marcadores aleatórios em torno de Madri e ative ou desative o agrupamento.
Heatmap	Adicione um mapa de calor de 5000 pontos ponderados e altere seu raio, gradiente e opacidade.
Drawing	Adicione um marcador arrastável, uma polilinha editável, um polígono editável com um furo, um retângulo editável e uma linha geodésica de Madri a Nova York.
Rotation	Habilite a rotação, defina o bearing, gire com Ctrl+arrastar ou com arraste do botão direito, e redefina o mapa com o norte para cima.
Export	Exporte as sobreposições para GeoJSON, GPX ou KML, e salve o mapa como PNG (BMP antes do Delphi 2009).

O log mostra OnMarkerDragEnd, OnShapeEdited, OnLayerChanged, OnClusterClick e OnBearingChanged.

Componentes e units

TsgcVCLMap, a lista de provedores de sgcMaps_Providers (sgcMapGetProviderInfo), TsgcMapGPX, sgcMapSaveGeoJSONFile, sgcMapSaveKMLFile e sgcMapDistance. Units: sgcMaps_Classes, sgcMaps_VCL, sgcMaps_Providers, sgcMaps_Geometry, sgcMaps_GPX, sgcMaps_KML e sgcMaps_GeoJSON.

Como executar

Abra **sgcMapsFeaturesDemo.dpr** no IDE, depois compile e execute. O demo encontra as units do sgcMaps pelo library path definido pelo instalador sgcUI; com uma instalação manual, adicione primeiro a pasta Source ao library path.

Dados e chaves

- Ao iniciar, o demo adiciona uma camada base para cada provedor que não precisa de chave: OpenStreetMap, OpenTopoMap, as camadas Esri e CARTO Light e Dark.
- MapTiler, Thunderforest e Stadia precisam de uma chave de API obtida em seus sites. Digite-a na aba Layers e pressione Apply key.

- Cada provedor tem seu próprio cache de tiles em **%TEMP%\sgcMapsFeaturesDemo**.
- Os arquivos exportados são gravados onde você escolher na caixa de diálogo de salvar.

03 Services

Pasta	Projeto	Framework	Versão mínima
Demos\Maps\03.Services	sgcMapsServicesDemo.dpr	VCL	Delphi XE8

O que mostra

Todos os serviços online em um único formulário. Cada aba tem uma lista de provedores e uma caixa de chave de API, e o resultado é desenhado no mapa.

Aba	O que você pode fazer
Search	Sugestões de lugares enquanto você digita (Photon, Nominatim, Google, HERE, Mapbox ou Geoapify) e geocodificação de um endereço.
Reverse	Clique no mapa para obter o endereço daquele ponto.
Directions	Uma rota entre dois endereços ou entre pontos de passagem clicados no mapa (arrastáveis), com um modo de viagem, alternativas, e OSRM, OpenRouteService, Google Routes, HERE, Azure Maps, Mapbox ou Geoapify.
Elevation	O perfil de elevação da rota, de Open-Elevation, OpenTopoData ou Google.
Time zone	O fuso horário no centro do mapa, de TimeAPI, GeoNames, Google ou Azure Maps.
Location	Simule um trajeto de GPS ao longo da rota, faça o mapa seguir a posição, ou use a localização do dispositivo.

Componentes e units

TsgcMapPlaces, TsgcMapGeocoder, TsgcMapDirections, TsgcMapRouteCalculator, TsgcMapElevation, TsgcMapTimeZone e TsgcMapLocation, criados em código em um formulário com um TsgcVCLMap.

Como executar

Abra **sgcMapsServicesDemo.dpr** no IDE, depois compile e execute. O demo encontra as units do sgcMaps pelo library path definido pelo instalador sgcUI; com uma instalação manual, adicione primeiro a pasta Source ao library path.

Dados e chaves

- Os provedores padrão não precisam de chave: Photon, Nominatim, o servidor de demonstração do OSRM, OpenTopoData e TimeAPI.
- Google, HERE, Azure Maps, Mapbox, OpenRouteService e Geoapify precisam de uma chave obtida em seus consoles de desenvolvedor. Para o GeoNames, digite seu nome de usuário do GeoNames como chave.
- O demo define seu próprio UserAgent em cada serviço, como os servidores públicos pedem, e mostra seus limites no formulário: uma requisição por segundo e nenhuma requisição em massa. Use seu próprio servidor ou um provedor comercial em produção.

- Os tiles vêm do OpenStreetMap e ficam em cache na pasta **tilecache** ao lado do executável.
- Os erros dos serviços são gravados no log. Quando a localização do dispositivo não está disponível, o demo informa isso e desmarca a opção.

04 Offline Services

Pasta	Projeto	Framework	Versão mínima
Demos\Maps\04.OfflineServices	sgcMapsOfflineDemo.dpr	VCL	Delphi XE7 (FireDAC SQLite)

O que mostra

Busca e roteamento sem conexão de rede. O demo constrói seus arquivos de dados a partir de downloads gratuitos, abre-os novamente na próxima inicialização e responde a todas as requisições localmente. O mapa começa em Andorra.

- **Offline tiles** com **Folder...** ou **MBTiles...** define um mapa base local. Sem isso, ou quando o arquivo não existe, o demo usa os tiles online do OpenStreetMap e informa no log.
- **Build from GeoNames...** constrói um índice do geocodificador a partir de um dump do GeoNames, e **Open index...** abre um já existente.
- **Build from .osm.pbf...** constrói um grafo de rotas a partir de um extrato do OpenStreetMap, e **Open graph...** abre um já existente.
- **Search** encontra um lugar no índice, e um clique no mapa faz a geocodificação reversa daquele ponto.
- Cliques no mapa adicionam pontos de passagem da rota, que podem ser arrastados. A rota é calculada para o perfil Car, Bike ou Foot e seus passos são listados.

Componentes e units

TsgcMapOfflineGeocoder, TsgcMapGeoIndexBuilder, TsgcMapOfflineRouter, TsgcMapRouteGraphBuilder e TsgcMapRouteCalculator, que chama o roteador offline a partir de seu evento OnCalculateRoute.

Como executar

Abra **sgcMapsOfflineDemo.dpr** no IDE, depois compile e execute. O demo encontra as units do sgcMaps pelo library path definido pelo instalador sgcUI; com uma instalação manual, adicione primeiro a pasta Source ao library path.

Dados e chaves

- Nenhuma chave de API é necessária.
- Um dump do GeoNames de https://download.geonames.org/export/dump/, como ES.zip ou cities15000.zip. Quando admin1CodesASCII.txt e countryInfo.txt estão na mesma pasta, o demo os usa também. O índice é gravado ao lado do dump com a extensão .sqlite.
- Um extrato do OpenStreetMap, como andorra-latest.osm.pbf de um serviço de download como o Geofabrik. O grafo é gravado ao lado dele com a extensão .sgcgraph.
- As construções rodam em uma thread com uma barra de progresso, e o formulário não pode ser fechado até terminarem.
- Os caminhos do índice, do grafo e dos tiles são salvos em um arquivo .ini ao lado do executável e reabertos na inicialização.
- As ferramentas constroem os mesmos arquivos pela linha de comando.

05 Static Render

Pasta	Projeto	Fra-mework	Versão mínima
Demos\Maps\05.StaticRender	sgcMapsStaticRender.dpr	Console	Delphi 7 (tiles online e PNG a partir do Delphi 2009, rotas a partir do Delphi XE2, GeoJSON a partir do Delphi XE6)

O que mostra

Uma aplicação de console que cria um TsgcVCLMap sem janela pai, carrega os tiles de uma visualização e a salva como imagem. É a base para um serviço de mapas estáticos ou para imagens de mapa em relatórios. Veja Renderização de Mapas Estáticos.

Como executar

Compile o projeto e execute-o em um prompt de comando:

```
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]
```

Parâmetro	Significado
--center lat,lon	O centro da visualização. Obrigatório, a menos que --route seja informado, o que centraliza a visualização na rota.
--zoom z	O nível de zoom. O padrão é 13.
--size WxH	O tamanho da imagem em pixels. O padrão é 800x600.
--provider	osm (o padrão), opentopo, esri ou carto. Esses provedores não precisam de chave.
--out file	A imagem a gravar. Obrigatório. Uma extensão .bmp grava um arquivo BMP, qualquer outra extensão um arquivo PNG (um arquivo BMP antes do Delphi 2009).
--marker lat,lon	Adiciona um marcador. Pode ser repetido.
--geojson file	Desenha as feições de um arquivo GeoJSON (Delphi XE6 e posterior).
--route and --graph	Calcula uma rota pelos pontos de passagem com o roteador offline e o arquivo .sgcgraph informado, e a desenha (Delphi XE2 e posterior).

As coordenadas usam o ponto como separador decimal. O programa imprime o provedor, a visualização, o arquivo salvo e a atribuição, e termina com o código 0 em caso de sucesso, 1 para um parâmetro errado e 2 quando ocorre uma exceção.

```
sgcMapsStaticRender --center 48.8584,2.2945 --zoom 15 --size 1024x768 --provider osm --out eiffel.png
```

Dados e chaves

- Nenhuma chave de API é necessária.
- Os tiles são baixados quando não estão na pasta **tile_cache** ao lado do executável, portanto a primeira execução precisa de conexão de rede.
- A ferramenta RouteGraphBuilder constrói o arquivo .sgcgraph para --route.

06 FireMonkey

Pasta	Projeto	Framework	Versão mínima
Demos\Maps\06.FMX	sgcMapsFMXDemo.dpr	FireMonkey	Delphi XE6 (busca e roteamento a partir do Delphi XE8)

O que mostra

Um TsgcFMXMap criado em código, com quatro abas:

Aba	O que você pode fazer
Search	Sugestões de lugares enquanto você digita e **Route from here** até o lugar selecionado.
Location	**Locate me** e **Follow my location**.
Overlays	Agrupamento de marcadores com 500 marcadores, um mapa de calor, um marcador arrastável, um polígono editável e **Clear overlays**.
View	A camada base, o seletor de camadas, a rotação com dois dedos, o bearing, **Reset compass** e **Snapshot to PNG**, que salva sgcMapsFMX.png na pasta de documentos.

Componentes e units

TsgcFMXMap, TsgcMapLocation, TsgcMapPlaces com Photon e TsgcMapDirections com OSRM. O mapa é criado em código, portanto o demo roda sem o pacote de design.

Como executar

Abra **sgcMapsFMXDemo.dpr** no IDE, escolha a plataforma de destino, depois compile e execute. No Android a permissão de localização é solicitada em tempo de execução (Delphi 10.3 e posterior).

Dados e chaves

- Nenhuma chave de API é necessária. As camadas base são os provedores que não precisam de chave, e Places e Directions definem seu próprio UserAgent.
- O cache de tiles fica na pasta temporária do dispositivo, que é gravável em todas as plataformas.
- Quando o dispositivo não tem sensor de localização, quando nenhuma posição chega em cinco segundos, ou quando a permissão é negada, o demo usa uma posição simulada em Madri para que o restante do demo continue funcionando.

07 Vector Tiles

Pasta	Projeto	Framework	Versão mínima
Demos\Maps\07.VectorTiles	sgcMapsVectorDemo.dpr	VCL	Delphi XE2 (MBTiles local a partir do Delphi XE7)

O que mostra

Tiles vetoriais desenhados pelo TsgcVCLMap a partir de três fontes e com três estilos. Veja Tiles Vetoriais.

- Fontes: **OpenFreeMap** (online, sem chave), **MapTiler vector** (precisa de chave) e um arquivo **local .mbtiles** de tiles vetoriais.
- Estilos: o estilo integrado, o estilo de exemplo **Liberty**, ou qualquer arquivo JSON de estilo MapLibre.
- **Retina** desenha os tiles com 512 pixels, e o controle deslizante de bearing ou Ctrl+arrastar gira o mapa.
- A barra de status mostra o zoom, o número de rótulos desenhados e os erros de tiles informados por OnTileError.

Como executar

Abra **sgcMapsVectorDemo.dpr** no IDE, depois compile e execute. O demo encontra as units do sgcMaps pelo library path definido pelo instalador sgcUI; com uma instalação manual, adicione primeiro a pasta Source ao library path.

Dados e chaves

- O OpenFreeMap não precisa de chave. Ao iniciar, o demo lê a URL atual dos tiles no TileJSON do OpenFreeMap (Delphi XE8 e posterior) e recorre a uma URL fixa quando não consegue.
- O MapTiler precisa de uma chave obtida no site do MapTiler Cloud.
- Um arquivo local deve conter tiles vetoriais no esquema OpenMapTiles, o que o estilo integrado desenha.
- O estilo de exemplo Liberty é lido dos fixtures de QA da instalação. Quando o arquivo não é encontrado, essa opção fica desabilitada.

TsgcVCLMap

Unidade: sgcMaps_VCL
O controle de mapa VCL. Ele descende de TCustomControl e desenha os tiles e as sobreposições por conta própria, portanto não precisa de navegador, de WebView nem de JavaScript. No Lazarus a mesma classe é o controle de mapa LCL, que exibe apenas tiles de uma pasta porque não consegue baixar tiles (veja Lazarus).
O controle exibe as coleções do modelo de mapa. Adicione uma camada de tiles a **Layers**, defina a visualização com **SetCenter** e **Zoom**, e adicione marcadores e formas com os métodos Add ou diretamente em suas coleções. O usuário move o mapa com o mouse, aplica zoom com a roda ou com os botões de zoom, e gira o mapa com Ctrl+arrastar ou com arraste do botão direito quando **Options.RotationEnabled** é True.

Propriedades

Propriedade	Tipo	Padrão	Descrição
CenterLatitude	Double		Latitude do centro da visualização, em graus decimais (WGS 84).
CenterLongitude	Double		Longitude do centro da visualização, em graus decimais (WGS 84).
Zoom	Integer	13	Nível de zoom, limitado por Options.MinZoom e Options.MaxZoom.
Layers	TsgcMapLayers		Camadas de tiles, desenhadas na ordem da coleção. Veja as propriedades de camada abaixo.
Markers	TsgcMapMarkers		Marcadores: Latitude, Longitude, PopupText, TooltipText, Color, IconBitmap, AnchorX, AnchorY, Draggable, Visible, Tag e Data.
Polylines	TsgcMapPolylines		Linhas: AddPoint, Color, Width, Opacity, Dashed, Geodesic, Editable, Visible e Tag.
Polygons	TsgcMapPolygons		Polígonos preenchidos, com furos.
Rectangles	TsgcMapRectangles		Retângulos preenchidos, dados por suas bordas norte, oeste, sul e leste.
Circles	TsgcMapCircles		Círculos: CenterLat, CenterLon, RadiusMeters, FillColor, StrokeColor, StrokeWidth e FillOpacity.
Labels	TsgcMapLabels		Rótulos de texto ancorados a uma coordenada.
Heatmaps	TsgcMapHeatmaps		Mapas de calor de pontos ponderados, com os gradientes mhgDefault, mhgFire, mhgCool e mhgGrayscale.
Options	TsgcMapOptions		Opções da visualização: MinZoom, MaxZoom, TileSize, ShowZoomControls, ShowScaleBar, ShowAttribution, AttributionText, BackgroundColor, DragEnabled, ZoomEnabled, Bearing, RotationEnabled e ShowLayerSwitcher.

MarkerOptions	TsgcMapMarkerOptions		Tamanho, cor e fonte padrão dos marcadores, e Clustering (Enabled, GridSize, MinClusterSize, Color, FontColor, ZoomOnClick, MaxZoom).
ActivePopupMarker	TsgcMapMarker		O marcador cujo popup está aberto, ou nil. Somente leitura.
DrawnMarkerCount	Integer		Marcadores desenhados na última pintura. Somente leitura.
DrawnClusterCount	Integer		Clusters desenhados na última pintura. Somente leitura.
DrawnLabelCount	Integer		Rótulos desenhados na última pintura. Somente leitura.
LastRenderMissingTiles	Integer		Tiles que ainda não estavam disponíveis na última pintura. Somente leitura.

Propriedades de camada

Cada item de **Layers** é um TsgcMapLayer.

Propriedade	Tipo	Padrão	Descrição
Name	string		Nome exibido pelo seletor de camadas.
TileSourceType	TsgcMapTileSourceType	mtsFolder	mtsFolder (pasta z/x/y), mtsMBTiles, mtsHTTP, mtsVectorMBTiles ou mtsVectorHTTP.
TileSourcePath	string		Pasta ou arquivo MBTiles de uma camada offline.
URLTemplate	string		URL dos tiles de uma camada online, com os marcadores de posição {z}, {x} e {y}.
CacheFolder	string		Cache em disco de uma camada online. Vazio desativa o cache.
Provider	TsgcMapTileProvider	mtpCustom	Um provedor integrado. A camada baixa seus tiles sem nenhuma outra configuração; sgcMapApplyProvider também copia para a camada o tipo de fonte, o modelo de URL, o zoom máximo e a atribuição do provedor.
ApiKey	string		Chave dos provedores que precisam de uma (MapTiler, Thunderforest, Stadia).
Attribution	string		Texto de atribuição da camada.
Opacity	Double		Opacidade de 0 a 1.
Visible	Boolean	True	Mostra ou oculta a camada.
MinZoom, MaxZoom	Integer	0, 19	Faixa de zoom dos tiles da camada.

IsBase-Layer	Boolean	True	True para uma camada base, False para uma sobreposição desenhada por cima. Mostrar uma camada base com SetLayerVisible ou no seletor de camadas oculta as outras camadas base.
Retina	Boolean	False	Solicita tiles de alta resolução.
VectorSty-leFile	string		Estilo MapLibre ou Mapbox GL de uma camada vetorial.

Métodos

Método	Descrição
SetCenter(aLat, aLon: Double)	Move o centro da visualização.
ZoomIn, ZoomOut	Altera o zoom em um nível.
FitBounds(aNorthLat, aWestLon, aSouthLat, aEastLon: Double; aPaddingPx: Integer = 20)	Centraliza e aplica zoom na visualização para mostrar uma área.
FitToMarkers(aPaddingPx: Integer = 20)	Centraliza e aplica zoom na visualização para mostrar todos os marcadores.
FitToPolyline(aPolyline: TsgcMapPolyline; aPaddingPx: Integer = 20)	Centraliza e aplica zoom na visualização para mostrar uma polilinha, por exemplo uma rota.
AddMarker(aLat, aLon: Double; const aPopup: string = ''; const aTooltip: string = '')	Adiciona um marcador com texto de popup e dica de ferramenta opcionais.
ClearMarkers	Remove todos os marcadores.
AddPolyline: TsgcMapPolyline	Adiciona uma polilinha vazia; adicione seus pontos com AddPoint.
AddPolygon: TsgcMapPolygon	Adiciona um polígono vazio.
AddCircle(aLat, aLon, aRadiusMeters: Double): Tsgc-MapCircle	Adiciona um círculo.
AddRectangle(aNorthLat, aWestLon, aSouthLat, aEas-tLon: Double): TsgcMapRectangle	Adiciona um retângulo.
AddLabel(aLat, aLon: Double; const aText: string): TsgcMapLabel	Adiciona um rótulo de texto.
ShowPopup(aMarker: TsgcMapMarker), HidePopup	Abre ou fecha o popup de um marcador.
MarkerAtScreen(aX, aY): TsgcMapMarker	O marcador em um ponto do controle, ou nil.
ClusterAtScreen(aX, aY; out aCount: Integer; out aLat, aLon: Double): Boolean	True quando há um cluster em um ponto do controle, com seu tamanho e posição.
SetLayerVisible(aIndex: Integer; aVisible: Boolean)	Mostra ou oculta uma camada. Mostrar uma camada base oculta as outras camadas base, que é também o que o seletor de camadas faz.
RefreshTiles	Recarrega os tiles da visualização.
AttributionText: string	A atribuição das camadas visíveis.

RenderToBitmap(aBitmap: TBitmap; aIncludeControls: Boolean = False)	Desenha a visualização atual em um bitmap.
SaveToPNG, SaveToBMP(const aFileName: string; aWidth, aHeight: Integer; aIncludeControls: Boolean = False)	Renderiza a visualização no tamanho informado em um arquivo PNG ou BMP. Funciona sem janela pai.

Eventos

Evento	Descrição
OnMapClick	Um clique no mapa, com sua latitude e longitude.
OnMarkerClick	Um clique em um marcador.
OnMarkerDragStart, OnMarker-Drag, OnMarkerDragEnd	O usuário arrasta um marcador cuja propriedade Draggable é True.
OnShapeEdited	O usuário moveu um vértice de uma polilinha, polígono ou retângulo editável.
OnClusterClick	Um clique em um cluster, com seu tamanho e posição. Defina aHan-dled como True para tratar o clique você mesmo.
OnLayerChanged	O usuário escolheu outra camada no seletor de camadas.
OnZoomChanged	O nível de zoom mudou.
OnBearingChanged	O mapa foi girado.
OnTileError	Um tile não pôde ser carregado, com a camada, as coordenadas do tile e o erro.

Exemplo: Configurando o mapa

Um manipulador OnCreate que adiciona uma camada OpenStreetMap, centraliza a visualização e adiciona um marcador, uma linha e um 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;
```

Exemplo: Eventos e snapshots

```
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;
```

Veja Também

- TsgcFMXMap
- Início Rápido VCL
- TsgcMapRouteCalculator

TsgcFMXMap

Unidade: sgcMaps_FMX
O controle de mapa FireMonkey, disponível a partir do Delphi XE6. Ele descende do TControl do FireMonkey e compartilha o modelo de mapa com TsgcVCLMap: as mesmas Layers, Markers, Polylines, Polygons, Rectangles, Circles, Labels, Heatmaps, Options e MarkerOptions, os mesmos métodos e os mesmos eventos.

Diferenças em relação ao TsgcVCLMap

- Entrada por toque: um dedo move o mapa, um gesto de pinça aplica zoom e, com **Options.RotationEnabled**, dois dedos giram o mapa.
- **MarkerAtScreen** e **ClusterAtScreen** recebem coordenadas Single, o tipo de coordenada do FireMonkey.
- **RenderToBitmap** recebe um TBitmap do FMX, e o controle publica **HitTest** (True por padrão).
- Nas plataformas móveis a pasta da aplicação é somente leitura: coloque o **CacheFolder** da camada em uma pasta gravável, como TPath.GetCachePath, e salve os snapshots em TPath.GetDocumentsPath.

Propriedades

Propriedade	Tipo	Padrão	Descrição
CenterLatitude	Double		Latitude do centro da visualização, em graus decimais (WGS 84).
CenterLongitude	Double		Longitude do centro da visualização, em graus decimais (WGS 84).
Zoom	Integer	13	Nível de zoom, limitado por Options.MinZoom e Options.MaxZoom.
Layers	TsgcMapLayers		Camadas de tiles, desenhadas na ordem da coleção. Veja as propriedades de camada abaixo.
Markers, Polylines, Polygons, Rectangles, Circles, Labels, Heatmaps	coleções		As sobreposições, como em TsgcVCLMap.
Options, MarkerOptions	TsgcMapOptions, TsgcMapMarkerOptions		Opções de visualização, de marcadores e de agrupamento, como em TsgcVCLMap.
Canvas	TCanvas		O canvas FireMonkey do controle. Somente leitura.
ActivePopupMarker	TsgcMapMarker		O marcador cujo popup está aberto, ou nil. Somente leitura.
DrawnMarkerCount	Integer		Marcadores desenhados na última pintura. Somente leitura.
DrawnClusterCount	Integer		Clusters desenhados na última pintura. Somente leitura.
DrawnLabelCount	Integer		Rótulos desenhados na última pintura. Somente leitura.

LastRenderMissingTiles	Integer		Tiles que ainda não estavam disponíveis na última pintura. Somente leitura.

Métodos

Método	Descrição
SetCenter(aLat, aLon: Double)	Move o centro da visualização.
ZoomIn, ZoomOut	Altera o zoom em um nível.
FitBounds(aNorthLat, aWestLon, aSouthLat, aEastLon: Double; aPaddingPx: Integer = 20)	Centraliza e aplica zoom na visualização para mostrar uma área.
FitToMarkers(aPaddingPx: Integer = 20)	Centraliza e aplica zoom na visualização para mostrar todos os marcadores.
FitToPolyline(aPolyline: TsgcMapPolyline; aPaddingPx: Integer = 20)	Centraliza e aplica zoom na visualização para mostrar uma polilinha, por exemplo uma rota.
AddMarker(aLat, aLon: Double; const aPopup: string = ''; const aTooltip: string = '')	Adiciona um marcador com texto de popup e dica de ferramenta opcionais.
ClearMarkers	Remove todos os marcadores.
AddPolyline: TsgcMapPolyline	Adiciona uma polilinha vazia; adicione seus pontos com AddPoint.
AddPolygon: TsgcMapPolygon	Adiciona um polígono vazio.
AddCircle(aLat, aLon, aRadiusMeters: Double): TsgcMapCircle	Adiciona um círculo.
AddRectangle(aNorthLat, aWestLon, aSouthLat, aEastLon: Double): TsgcMapRectangle	Adiciona um retângulo.
AddLabel(aLat, aLon: Double; const aText: string): TsgcMapLabel	Adiciona um rótulo de texto.
ShowPopup(aMarker: TsgcMapMarker), HidePopup	Abre ou fecha o popup de um marcador.
MarkerAtScreen(aX, aY): TsgcMapMarker	O marcador em um ponto do controle, ou nil.
ClusterAtScreen(aX, aY; out aCount: Integer; out aLat, aLon: Double): Boolean	True quando há um cluster em um ponto do controle, com seu tamanho e posição.
SetLayerVisible(aIndex: Integer; aVisible: Boolean)	Mostra ou oculta uma camada. Mostrar uma camada base oculta as outras camadas base, que é também o que o seletor de camadas faz.
RefreshTiles	Recarrega os tiles da visualização.
AttributionText: string	A atribuição das camadas visíveis.
RenderToBitmap(aBitmap: TBitmap; aIncludeControls: Boolean = False)	Desenha a visualização atual em um bitmap.

SaveToPNG, SaveToBMP(const aFileName: string; aWidth, aHeight: Integer; aIncludeControls: Boolean = False)	Renderiza a visualização no tamanho informado em um arquivo PNG ou BMP. Funciona sem janela pai.

Eventos

Evento	Descrição
OnMapClick	Um clique no mapa, com sua latitude e longitude.
OnMarkerClick	Um clique em um marcador.
OnMarkerDragStart, OnMarker-Drag, OnMarkerDragEnd	O usuário arrasta um marcador cuja propriedade Draggable é True.
OnShapeEdited	O usuário moveu um vértice de uma polilinha, polígono ou retângulo editável.
OnClusterClick	Um clique em um cluster, com seu tamanho e posição. Defina aHan-dled como True para tratar o clique você mesmo.
OnLayerChanged	O usuário escolheu outra camada no seletor de camadas.
OnZoomChanged	O nível de zoom mudou.
OnBearingChanged	O mapa foi girado.
OnTileError	Um tile não pôde ser carregado, com a camada, as coordenadas do tile e o erro.

Exemplo: Configurando o mapa

Um manipulador OnCreate que adiciona uma camada OpenStreetMap com seu cache na pasta de cache da aplicação, ativa a rotação e o agrupamento, e salva um snapshot na pasta 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;
```

Veja Também

- TsgcVCLMap
- Início Rápido FMX
- TsgcMapLocation

Tiles, Camadas e Provedores

Um mapa é desenhado a partir de **tiles**: imagens quadradas, de 256 pixels de largura por padrão, que cobrem o mundo em cada nível de zoom. No zoom 0 um tile mostra o mundo inteiro, e cada nível seguinte dobra o número de tiles em cada direção. Um tile é endereçado por seu zoom, x e y, o esquema z/x/y usado pelo OpenStreetMap e por quase todos os servidores de tiles.
A coleção **Layers** de TsgcVCLMap e TsgcFMXMap diz de onde vêm os tiles. Cada item é um **TsgcMapLayer** (unit sgcMaps_Classes), e seu **TileSourceType** seleciona a fonte:

TileSourceType	Os tiles vêm de	Caminho ou URL
mtsFolder	Uma pasta z/x/y de arquivos .png, .jpg ou .jpeg no disco.	TileSourcePath
mtsMBTiles	Um arquivo MBTiles raster.	TileSourcePath
mtsHTTP	Um servidor de tiles, por meio de um modelo de URL.	URLTemplate
mtsVectorHTTP	Um servidor de tiles vetoriais. Veja Tiles Vetoriais.	URLTemplate
mtsVectorMBTiles	Um arquivo MBTiles vetorial. Veja Tiles Vetoriais.	TileSourcePath

As fontes de pasta e MBTiles funcionam sem conexão de rede; Tiles Offline e o Baixador de Tiles explica como prepará-las.

Provedores integrados

A propriedade **Provider** de uma camada nomeia um dos servidores de tiles integrados de **TsgcMapTileProvider**. Chame **sgcMapApplyProvider** (unit sgcMaps_Providers) depois de defini-la: ela define TileSourceType como mtsHTTP, copia para a camada o modelo de URL e o zoom máximo do provedor, e preenche Attribution quando está vazio. Uma camada cujo Provider não é mtpCustom baixa os tiles de seu provedor mesmo sem essa chamada, mas então MaxZoom mantém seu padrão de 19 e as propriedades da camada não mostram as configurações do provedor.

Provedor	Chave de API	Zoom máx.	Tiles retina
mtpOpenStreetMap	Não	19	Não
mtpOpenTopoMap	Não	17	Não
mtpEsriWorldImagery	Não	19	Não
mtpEsriWorldStreet	Não	19	Não
mtpCartoLight	Não	20	Sim
mtpCartoDark	Não	20	Sim
mtpMapTilerStreets	Sim	22	Sim
mtpMapTilerSatellite	Sim	20	Sim
mtpThunderforestCycle	Sim	22	Sim
mtpThunderforestTransport	Sim	22	Sim
mtpStadiaAlidadeSmooth	Sim	20	Sim

mtpStadiaAlidadeSmoothDark	Sim	20	Sim

Defina a chave dos provedores que precisam de uma na propriedade **ApiKey** da camada. Servidores públicos têm políticas de uso: tile.openstreetmap.org, por exemplo, proíbe uso intenso e downloads em massa, portanto uma aplicação em produção deve usar um provedor comercial ou seu próprio servidor de tiles.

Modelos de URL

Para qualquer outro servidor, defina TileSourceType como mtsHTTP e escreva seu **URLTemplate**. Estes marcadores de posição são substituídos em cada tile:

Marcador de posição	Substituído por
{z}, {x}, {y}	O zoom e a coluna e a linha do tile.
{s}	Uma letra de subdomínio, alternada por tile para que as requisições se distribuam entre os servidores. As letras são a, b e c, a menos que o provedor defina outras.
{r}	@2x quando a camada tem Retina definido como True, vazio caso contrário.
{apikey}	O ApiKey da camada.

A mesma expansão está disponível como **sgcMapExpandTileURL**. **sgcMapProviderNames** preenche uma lista de strings com os nomes dos provedores, e **sgcMapGetProviderInfo** retorna um registro **TsgcMapProviderInfo** com o nome, o modelo de URL, os subdomínios, a atribuição, o zoom máximo e se o provedor precisa de uma chave de API ou oferece suporte a tiles 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;
```

Várias camadas

Uma camada é uma **camada base** (IsBaseLayer True, o padrão) ou uma **sobreposição**. O mapa desenha a primeira camada base visível e depois cada sobreposição visível por cima dela, na ordem da coleção. Opacity define a transparência de uma camada, MinZoom e MaxZoom limitam os níveis de zoom em que ela é desenhada, e Retina pede ao provedor tiles com o dobro da resolução.
Defina **Options.ShowLayerSwitcher** como True para permitir que o usuário escolha as camadas no mapa. As camadas base se comportam como itens de rádio: mostrar uma com **SetLayerVisible** oculta as outras camadas base. O evento **OnLayerChanged** informa a camada que foi mostrada ou ocultada.

```
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;
```

Cache de tiles

Defina **CacheFolder** para manter em disco cada tile baixado, em uma árvore z/x/y. Tiles que já estão no cache são desenhados sem requisição ao servidor, de modo que um mapa que o usuário viu uma vez continua funcionando sem conexão. O mapa também mantém em memória os tiles recentes; **RefreshTiles** recarrega os tiles da visualização. Um tile que não pode ser carregado dispara **OnTileError** com a camada, o endereço do tile e o texto do erro, e **LastRenderMissingTiles** conta os tiles ausentes na última pintura.

Disponibilidade

Framework	Versão mínima
VCL, tiles de pasta	Delphi 7
VCL, tiles de um servidor (provedores, modelos de URL, cache)	Delphi 2009
VCL e FireMonkey, MBTiles	Delphi XE7 (FireDAC SQLite)
FireMonkey	Delphi XE6
Lazarus (LCL)	Somente tiles de pasta

Do Delphi 2009 ao XE7 os tiles são baixados com o Indy 10, e o https precisa das bibliotecas OpenSSL 1.0.2 (libeay32.dll e ssleay32.dll) ao lado do executável. A partir do Delphi XE8 o mapa usa THTTPClient com o suporte a TLS do sistema operacional e não precisa de nenhuma DLL. No Delphi 7, no Delphi 2007 e no Lazarus não há download de tiles, portanto use uma pasta de tiles.

Veja Também

- TsgcVCLMap
- Tiles Vetoriais
- Tiles Offline e o Baixador de Tiles

Tiles Vetoriais

Um **tile vetorial** armazena estradas, edifícios, água e nomes de lugares como geometria em vez de pixels, no formato Mapbox Vector Tile (MVT). O sgcMaps decodifica os tiles e os desenha por conta própria, seguindo um **estilo**: um JSON de estilo MapLibre ou Mapbox GL que diz quais feições desenhar, com quais cores e larguras, em quais níveis de zoom. Um mesmo conjunto de tiles vetoriais pode, portanto, parecer claro, escuro ou minimalista conforme o estilo, e os tiles permanecem nítidos em qualquer densidade de pixels.

Uma camada vetorial

Adicione uma camada ao mapa com um dos tipos de fonte vetorial:

- **mtsVectorHTTP**: tiles de um servidor. URLTemplate usa os mesmos marcadores de posição dos tiles raster, por exemplo https://api.maptiler.com/tiles/v3/{z}/{x}/{y}.pbf?key={apikey}, e ApiKey preenche {apikey}.
- **mtsVectorMBTiles**: tiles de um arquivo MBTiles vetorial local definido em TileSourcePath, como um arquivo feito com tilemaker ou planetiler.

VectorStyleFile é o caminho de um arquivo JSON de estilo salvo como UTF-8. Quando está vazio, o mapa usa o estilo integrado, um estilo claro para o esquema **OpenMapTiles** (água, cobertura do solo, uso do solo, parques, edifícios, estradas, limites, cursos de água e os nomes de lugares, estradas e águas). Esse esquema é o usado por MapTiler, OpenFreeMap e a maioria dos tiles vetoriais do OpenStreetMap. Com **Retina** definido como True os tiles são renderizados com 512 pixels em vez de 256, para telas de alta DPI. Os rótulos das camadas de símbolo do estilo são coletados por tile e desenhados na vertical pelo mapa, também quando o mapa está girado.

```
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;
```

A fonte de tiles vetoriais

A camada cria um **TsgcMapTileSource_MVT** (unit sgcMaps_TileSource_MVT). Você também pode criar um por conta própria para renderizar tiles fora de um mapa:

Membro	Descrição
SourceKind	mvsHTTP ou mvsMBTiles.
URLTemplate, ApiKey, UserAgent	O servidor de mvsHTTP.

Path	O arquivo .mbtiles de mvsMBTiles.
StyleJSON, StyleFile	O estilo. StyleJSON tem precedência sobre StyleFile, e com ambos vazios Open carrega o estilo integrado.
TileSize	256 (padrão) ou 512 pixels.
MaxDataZoom	O maior zoom armazenado nos dados, 14 por padrão. Níveis de zoom mais profundos são desenhados a partir dos tiles desse nível.
CacheSize	O número de tiles renderizados mantidos em memória, 256 por padrão.
Open, Close	Open cria o leitor e carrega o estilo; ele gera uma exceção para um estilo inválido.
GetTile	Retorna o tile renderizado como uma imagem PNG.
GetTileLabels	Retorna os rótulos de um tile, com sua posição.
ClearCache	Descarta os tiles renderizados. Chame-o depois de alterar o estilo ou TileSize.
Style	O TsgcMapVectorStyle em uso, válido depois de Open.

Decodificação e estilos

A unit **sgcMaps_MVT** reúne os blocos de construção. **TsgcMapVectorTile** decodifica um tile a partir de bytes, de um stream ou de um arquivo, bruto ou compactado com gzip ou zlib, em camadas (**TsgcMapVectorLayer**) e feições (**TsgcMapVectorFeature**) com seu tipo de geometria, tags e geometria. **TsgcMapVectorStyle** carrega um estilo com LoadFromJSON ou LoadFromFile, ou o integrado com 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;
```

O suporte a estilos cobre os tipos de camada background, fill, line e symbol; camadas raster, fill-extrusion, circle, heatmap e hillshade são ignoradas. Filtros, funções de zoom e as expressões comuns (get, has, match, case, step, interpolate, coalesce e os operadores de comparação) são avaliados. Ícones, sprites, padrões, halos de texto e glifos não são usados.

Disponibilidade

Framework	Versão mínima
VCL e FireMonkey, tiles vetoriais de um servidor	Delphi XE2 (FireMonkey a partir do XE6)

VCL e FireMonkey, MBTiles vetorial	Delphi XE7 (FireDAC SQLite)
Lazarus (LCL)	Não disponível

Veja Também

- Tiles, Camadas e Provedores
- TsgcVCLMap
- TsgcFMXMap

Marcadores e Agrupamento

Um **marcador** fixa um ponto do mapa. A coleção **Markers** de TsgcVCLMap e TsgcFMXMap guarda itens **TsgcMapMarker**, e **AddMarker** adiciona um com um texto de popup e uma dica de ferramenta em uma única chamada.

Propriedades do marcador

Propriedade	Descrição
Latitude, Longitude	A posição do marcador.
PopupText	O texto do popup aberto por ShowPopup, geralmente a partir de OnMarkerClick. HidePopup o fecha e ActivePopupMarker é o marcador cujo popup está aberto.
TooltipText	O texto exibido enquanto o mouse está sobre o marcador.
Color	A cor do pino padrão. MarkerOptions.DefaultColor, DefaultWidth e DefaultHeight definem o pino de todos os marcadores.
IconBitmap	Um ícone personalizado, carregado com LoadFromFile ou LoadFromStream a partir de uma imagem PNG (Delphi 2009 e posterior, e Lazarus), JPEG ou BMP. Ele é armazenado no formulário, de modo que um ícone definido em tempo de design não precisa de arquivo em tempo de execução.
AnchorX, AnchorY	O ponto do ícone que toca a coordenada, como fração de sua largura e altura. O padrão, 0.5 e 1, é o centro inferior.
Draggable	Permite que o usuário mova o marcador. OnMarkerDragStart, OnMarkerDrag e OnMarkerDragEnd acompanham o arraste.
Visible, Tag, Data	Visibilidade, e um inteiro e uma string para seus próprios dados.

MarkerAtScreen retorna o marcador em um ponto do controle, **FitToMarkers** aplica zoom e centraliza a visualização em todos os marcadores, e **ClearMarkers** remove 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;
```

Agrupamento

Milhares de marcadores deixam um mapa lento e ilegível quando ele está reduzido. Com **MarkerOptions.Clustering.Enabled** definido como True, marcadores próximos são fundidos em um círculo que mostra quantos marcadores ele contém. A tela é dividida em células quadradas de **GridSize** pixels, e uma célula com pelo menos **MinClusterSize** marcadores vira um cluster desenhado no centro de seus marcadores. Acima de **MaxZoom** (18 por padrão) os marcadores nunca são agrupados. **Color** e **FontColor** definem a aparência dos círculos.
Um clique em um cluster dispara **OnClusterClick** com o número de marcadores e a posição do cluster. Quando o manipulador deixa aHandled como False e **ZoomOnClick** é True, o padrão, o mapa então aproxima até que os marcadores do cluster se separem. **ClusterAtScreen** encontra o cluster em um ponto do controle, e **DrawnClusterCount** e **DrawnMarkerCount** informam o que a última pintura desenhou.

```
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;
```

Envolva adições em massa em Markers.BeginUpdate e Markers.EndUpdate, para que o mapa seja repintado uma vez em vez de uma vez por marcador. A própria função de agrupamento, **sgcMapClusterPoints**, é pública em sgcMaps_Geometry para seus próprios pontos de tela.

Disponibilidade

Framework	Versão mínima
VCL	Delphi 7
FireMonkey	Delphi XE6
Lazarus (LCL)	Sim

Veja Também

- TsgcVCLMap
- Formas e Rótulos
- Mapas de Calor

Formas e Rótulos

Além dos marcadores, TsgcVCLMap e TsgcFMXMap desenham polilinhas, polígonos, retângulos, círculos e rótulos de texto sobre os tiles. Cada tipo tem sua própria coleção e um método Add no controle que cria um item e o retorna.

Coleção	Método Add	Item	Propriedades principais
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

Todo item também tem Visible, e as formas têm um Tag para seu uso próprio.

Polilinhas

Uma polilinha une seus pontos com retas no mapa. Com **Dashed** ela é desenhada tracejada, e com **Geodesic** cada segmento segue o círculo máximo, o caminho mais curto no globo, que é a escolha certa para voos e longas distâncias. **FitToPolyline** ajusta a visualização a uma polilinha, e as rotas dos componentes de direções e de roteamento também são adicionadas como polilinhas.

Polígonos com furos

Um polígono é preenchido dentro de seu anel externo. **Holes** adiciona anéis internos que ficam vazios: **AddHole** retorna o índice de um novo furo, e **AddPoint** adiciona um vértice a ele. Polígonos GeoJSON e KML com furos são importados da mesma forma.

```
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;
```

Edição

Defina **Editable** como True em uma polilinha, um polígono ou um retângulo para permitir que o usuário o remodele com o mouse: um vértice é movido arrastando-o, a alça no meio de um segmento insere um novo vértice, e um clique com o botão direito em um vértice o exclui. Quando uma edição termina, o mapa dispara **OnShapeEdited** com o item editado. Círculos e rótulos não são editáveis.

Disponibilidade

Framework	Versão mínima
VCL	Delphi 7
FireMonkey	Delphi XE6
Lazarus (LCL)	Sim

Veja Também

- TsgcVCLMap
- Marcadores e Agrupamento
- Funções de Geometria
- GeoJSON, GPX e KML

Mapas de Calor

Um **mapa de calor** mostra onde pontos ponderados se concentram: leituras de sensores, vendas, incidentes ou posições de GPS. Cada ponto espalha seu peso sobre um círculo, os pesos se somam e o total é colorido do frio ao quente.
A coleção **Heatmaps** de TsgcVCLMap e TsgcFMXMap guarda itens **TsgcMapHeatmap**. Adicione os pontos com **AddPoint**(latitude, longitude, peso); o peso é 1 quando omitido. Count, Points e Clear dão acesso aos pontos, e um registro **TsgcMapHeatmapPoint** guarda um deles.

Propriedade	Padrão	Descrição
Radius	25	O raio de influência de cada ponto, em pixels.
Opacity	0.6	A opacidade da cor mais quente.
MaxIntensity	0	A intensidade desenhada com a cor mais quente. 0 escala as cores até a maior intensidade desenhada; um valor fixo mantém as cores comparáveis entre visualizações.
Gradient	mhgDefault	mhgDefault (azul, ciano, lima, amarelo, vermelho), mhgFire, mhgCool ou mhgGrayscale.
Visible	True	Mostra ou oculta o 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;
```

O mapa de calor é desenhado em pixels de tela, de modo que sua forma acompanha o zoom: aproximar separa as manchas. A função de renderização, **sgcMapRenderHeatmap** em sgcMaps_Geometry, também é pública e preenche um buffer ARGB a partir de seus próprios pontos de tela.

Disponibilidade

Framework	Versão mínima
VCL	Delphi 7
FireMonkey	Delphi XE6
Lazarus (LCL)	Sim

Veja Também

- TsgcVCLMap
- Marcadores e Agrupamento

GeoJSON, GPX e KML

O sgcMaps lê e grava os três formatos de intercâmbio mais comuns de dados geográficos. As funções de importação adicionam itens às coleções de um mapa, e as funções de exportação gravam os itens de um mapa, de modo que qualquer coleção de TsgcVCLMap ou TsgcFMXMap pode ser passada.

Formato	Unit	Importação	Exportação
GeoJSON (RFC 7946)	sgcMaps_GeoJSON	Point e MultiPoint para marcadores, LineString e MultiLineString para polilinhas, Polygon e MultiPolygon para polígonos com furos, GeometryCollection, Feature e FeatureCollection.	Uma FeatureCollection com propriedades simplestyle. Retângulos são gravados como polígonos.
GPX	sgcMaps_GPX	Waypoints para marcadores, tracks e routes para polilinhas.	Marcadores para waypoints e polilinhas para tracks. Polígonos e retângulos são gravados como tracks fechados quando ExportPolygonsAsTracks é True.
KML	sgcMaps_KML	Point para marcadores, LineString para polilinhas, Polygon com seus limites internos para polígonos, com as cores dos estilos inline e compartilhados.	Um Placemark por item com um estilo inline. Retângulos são gravados como polígonos.

Cada formato tem uma classe (**TsgcMapGeoJSON**, **TsgcMapGPX**, **TsgcMapKML**) com LoadFromString, LoadFromFile, AsText, SaveToStream e SaveToFile, e as mesmas operações como funções simples: **sgcMapLoadGeoJSON**, **sgcMapLoadGeoJSONFile**, **sgcMapGeoJSONAsText**, **sgcMapSaveGeoJSONFile** e as funções GPX e KML correspondentes. Uma coleção passada como nil é ignorada.

```
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;
```

Estatísticas de trilhas GPX

TsgcMapGPX guarda estatísticas de cada track do último arquivo que carregou. **TrackStatsCount** e **TrackStats** retornam um registro **TsgcMapGPXTrackStats** com o nome, o número de pontos, a distância em metros, a elevação mínima e máxima, e a subida e a descida totais. Com **ColorByElevation** definido como True cada track é dividido em trechos coloridos por elevação, de LowColor passando por MidColor até HighColor de **ElevationGradient** (um 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;
```

Disponibilidade

Framework	Versão mínima
GPX e KML, VCL e FireMonkey	Delphi 7 (FireMonkey a partir do XE6)
GeoJSON, VCL e FireMonkey	Delphi XE6 (System.JSON)
Lazarus (LCL)	Não disponível

GPX e KML são analisados com TXMLDocument: MSXML no Windows e OmniXML nas outras plataformas a partir do Delphi XE7.

Veja Também

- Formas e Rótulos
- Marcadores e Agrupamento
- TsgcVCLMap

Funções de Geometria

A unit **sgcMaps_Geometry** reúne funções que trabalham com coordenadas, sem nenhum controle de mapa envolvido, de modo que servem a um serviço ou a uma aplicação de console tanto quanto a um formulário. As coordenadas são graus WGS84, as distâncias são metros em uma esfera de raio médio da Terra (SGC_MAP_EARTH_RADIUS, 6371008.8 m), e os bearings são graus no sentido horário a partir do norte verdadeiro, de 0 a 360.

Função	Retorna
sgcMapDistance	A distância de círculo máximo entre dois pontos, em metros.
sgcMapInitialBearing	O bearing inicial de um ponto para outro.
sgcMapDestinationPoint	O ponto alcançado a partir de um ponto inicial com um bearing e uma distância.
sgcMapGeodesicInterpolate	Os pontos do círculo máximo entre dois pontos, divididos em um número de segmentos.
sgcMapPointInPolygon	True quando um ponto está dentro de um anel.
sgcMapPointInPolygonWithHoles	True quando um ponto está dentro de um anel externo e fora de todos os seus furos.
sgcMapSimplify	Uma linha com menos pontos (Douglas-Peucker), dentro de uma tolerância em metros.
sgcMapNearestPointOnPolyline	A distância de um ponto a uma linha, com o ponto mais próximo e o índice de seu segmento.

Arrays de pontos usam **TsgcMapCoordArray**, um array do registro **TsgcMapCoord**, e **sgcMapCoord** constrói uma 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;
```

Limites e cálculos de tiles

FitBounds dos controles de mapa recebe uma caixa (norte, oeste, sul, leste) e a mostra com o maior zoom que cabe, mantendo uma margem em pixels; uma longitude oeste maior que a leste cruza o antimeridiano. FitToMarkers e FitToPolyline fazem o mesmo para os marcadores e para uma polilinha. A unit sgcMaps_Classes acrescenta os cálculos de tiles da projeção Web Mercator: **sgcMapLatLonToTileF** e **sgcMapTileFToLatLon** convertem entre coordenadas e números fracionários de tiles, **sgcMapLatLonToPixel** e **sgcMapPixelToLatLon** entre coordenadas e pixels do mundo, e **sgcMapTileCount** dá o número de tiles por lado em um 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;
```

Disponibilidade

Framework	Versão mínima
VCL, FireMonkey, console e serviços	Delphi 7
Lazarus	Sim

Veja Também

- Formas e Rótulos
- Marcadores e Agrupamento
- TsgcVCLMap

Renderização de Mapas Estáticos

Um controle de mapa pode renderizar sem janela. Um TsgcVCLMap criado sem pai pinta em um bitmap sob demanda, o que transforma uma aplicação de console, um servidor web ou um serviço do Windows em um renderizador de mapas estáticos: miniaturas para um relatório, um mapa anexado a um e-mail ou uma imagem servida por HTTP.

Métodos de renderização

Método	Descrição
RenderToBitmap(aBitmap, aInclude-Controls)	Pinta o mapa em um bitmap do seu próprio tamanho. Um bitmap vazio recebe o tamanho do controle.
SaveToPNG(aFileName, aWidth, aHeight, aIncludeControls)	Grava um arquivo PNG do tamanho informado.
SaveToBMP(aFileName, aWidth, aHeight, aIncludeControls)	Grava um arquivo BMP do tamanho informado.

A imagem mostra os tiles, as sobreposições, a barra de escala e a atribuição. Os botões de zoom, o seletor de camadas e a bússola ficam de fora, a menos que aIncludeControls seja True. Os tiles são lidos durante a renderização, e os tiles de um servidor são baixados nesse momento, portanto a chamada bloqueia até que cheguem. Depois, **LastRenderMissingTiles** informa quantos tiles não puderam ser carregados, e **AttributionText** retorna o texto de direitos autorais a exibir ao lado da imagem.

```
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.
```

Em um formulário, os mesmos métodos exportam a visualização que o usuário está vendo:

```
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;
```

O rasterizador

Os tiles vetoriais são desenhados com **TsgcMapRasterizer** (unit sgcMaps_Raster), um rasterizador de software com suavização que não depende de VCL nem de FireMonkey, e você pode usá-lo para desenhar suas próprias imagens. Ele preenche e traça caminhos (BeginPath, MoveTo, LineTo, ClosePath, FillPath com a regra non-zero ou even-odd, StrokePath com junções, pontas e tracejados), recorta a um retângulo e salva o resultado com SaveToBMPFile e SaveToPNGFile. As cores são valores $AARRGGBB diretos.

```
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;
```

Disponibilidade

Framework	Versão mínima
VCL, saída BMP	Delphi 7
VCL, saída PNG	Delphi 2009
FireMonkey	Delphi XE6
Lazarus (LCL)	Sim, PNG e BMP

Tiles de um servidor exigem o Delphi 2009 ou posterior e não estão disponíveis no Lazarus, onde o controle desenha apenas tiles de pasta; veja Tiles, Camadas e Provedores. O rasterizador comprime arquivos PNG com System.ZLib a partir do Delphi XE2 e os grava sem compressão em versões mais antigas.

Veja Também

- TsgcVCLMap
- TsgcFMXMap
- Tiles, Camadas e Provedores

Tiles Offline e o Baixador de Tiles

Um mapa que precisa funcionar sem conexão de rede, em um veículo, uma fábrica ou um navio, desenha seus tiles a partir de arquivos locais. O sgcMaps lê dois tipos de pacote de tiles offline:

- Uma **pasta** (TileSourceType mtsFolder): TileSourcePath contém uma subpasta por zoom, uma subpasta por coluna e um arquivo por tile, chamado z\x\y.png, .jpg ou .jpeg. Esse é o layout gravado por TsgcMapTileDownloader, pelo cache em disco de uma camada e por muitas ferramentas de tiles.
- Um arquivo **MBTiles** (TileSourceType mtsMBTiles): um único arquivo SQLite que guarda todos os tiles. É fácil de copiar e é produzido por ferramentas como MapTiler, TileMill e gdal. Os arquivos MBTiles vetoriais são descritos em Tiles Vetoriais.

```
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;
```

Baixando um pacote de tiles

TsgcMapTileDownloader (unit sgcMaps_TileDownloader) preenche uma pasta com todos os tiles de uma área e uma faixa de níveis de zoom.

Membro	Descrição
URLTemplate	O servidor de tiles, com os marcadores de posição {z} {x} {y}. Não tem padrão de propósito.
OutputPath	A pasta raiz da árvore z/x/y.
UserAgent	O cabeçalho User-Agent enviado com cada requisição.
DelayMS	A pausa entre dois downloads, 250 ms por padrão.
Timeout	O timeout de cada requisição, 15000 ms por padrão.
CountTiles	O número de tiles de uma área e faixa de zoom, para verificar o tamanho antes de baixar.
GetTileRange	A primeira e a última coluna e linha de tiles de uma área em um zoom.
Download	Baixa os tiles. Os tiles que já estão em disco são ignorados, de modo que um download interrompido pode ser retomado.

OnProgress	Informa o total de tiles, os baixados, os ignorados e os que falharam, e o zoom atual. Defina aCancel como True para parar.

Download é síncrono: bloqueia a thread que o chama e dispara OnProgress nessa mesma thread. Chame-o a partir de uma thread de trabalho para manter a interface do usuário responsiva. Ele gera uma exceção quando URLTemplate ou OutputPath está vazio.

```
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;
```

Respeite as políticas de uso dos tiles. A maioria dos servidores de tiles públicos, inclusive tile.openstreetmap.org, proíbe downloads em massa e pré-carga offline. Aponte o baixador para seu próprio servidor de tiles ou para um provedor cujos termos o permitam, e mantenha DelayMS acima de zero. O número de tiles quadruplica a cada nível de zoom, portanto verifique CountTiles antes de baixar níveis profundos.

Disponibilidade

Framework	Versão mínima
Tiles de pasta	Delphi 7, e Lazarus
MBTiles	Delphi XE7 (FireDAC SQLite)
TsgcMapTileDownloader	Delphi 2009 (Indy 10; THTTPClient a partir do Delphi XE8)
Lazarus (LCL), MBTiles e o baixador	Não disponível

Veja Também

- Tiles, Camadas e Provedores
- Tiles Vetoriais
- Instalação

Leitor de OpenStreetMap

Os dados do OpenStreetMap são distribuídos como arquivos **.osm.pbf** (Protocol Buffers, compactos e rápidos) e como arquivos XML **.osm**. Extratos por país ou região são publicados pelo Geofabrik e outros espelhos. **TsgcMapOSMReader** (unit sgcMaps_OSM) lê ambos os formatos como um stream: nunca constrói o documento inteiro em memória, portanto lê arquivos de qualquer tamanho e entrega cada elemento por meio de um evento enquanto o arquivo é lido.

Membro	Descrição
ReadFile, ReadStream	Lê um arquivo ou um stream. O formato é detectado pelo conteúdo; DetectFormat o retorna sem ler.
ReadNodes, ReadWays, ReadRelations	Quais elementos entregar, todos True por padrão. Desligar o que você não precisa torna a leitura mais rápida.
SkipNodeTags	Ignora as tags dos nós, para leitores que só precisam das coordenadas.
OnNode	Um nó com seu id, latitude, longitude e tags.
OnWay	Uma via com seu id, os ids de seus nós e suas tags.
OnRelation	Uma relação com seu id, seus membros (TsgcMapOSMMember: tipo, referência e papel) e suas tags.
OnFilterWay	Chamado antes de uma via ser entregue. Defina aAccept como False para ignorá-la.
OnProgress	Os bytes lidos e o tamanho total.

As tags chegam em um objeto **TsgcMapOSMTags** com Count, Keys, Values, Value, Has e IndexOf. O objeto de tags e os arrays passados aos eventos são reutilizados para o próximo elemento, portanto copie o que precisar dentro do manipulador.

```
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;
```

Blocos PBF compactados com zlib ou armazenados sem compressão são suportados. Blocos compactados com lzma, lz4, zstd ou bzip2, e arquivos que exigem recursos diferentes de OsmSchema-V0.6 e DenseNodes, geram **EsgcMapOSMError**. O leitor é a entrada de TsgcMapRouteGraphBuilder, veja Grafos de Rota e Roteamento Offline.

Disponibilidade

Framework	Versão mínima
VCL, FireMonkey, console e serviços	Delphi XE2 (System.ZLib)
Lazarus	Não disponível

Veja Também

- Grafos de Rota e Roteamento Offline
- TsgcMapOfflineRouter

Índice de Geocodificação Offline

A geocodificação sem conexão de rede precisa de um banco de dados local de nomes de lugares. O sgcMaps constrói um a partir dos dumps gratuitos do **GeoNames** e o consulta com SQLite, em duas etapas:

1. TsgcMapGeoIndexBuilder lê um dump do GeoNames (cities500.zip, cities1000.zip, um arquivo de país como ES.zip, allCountries.zip ou o arquivo .txt extraído), opcionalmente com admin1CodesASCII.txt e countryInfo.txt para nomes de regiões e países, e grava um banco SQLite indexado. FeatureClasses seleciona os tipos de lugares a manter, P para lugares povoados e A para áreas administrativas.
2. TsgcMapOfflineGeocoder abre esse banco e responde a Geocode (um nome de lugar, opcionalmente seguido de um país), ReverseGeocode e ReverseGeocodeCountry.

O índice é construído uma vez, em uma máquina de desenvolvimento ou no momento da instalação, e distribuído com a aplicação. Os resultados usam a mesma classe **TsgcMapGeocodingResults** do TsgcMapGeocoder online, de modo que o código que os exibe serve para ambos.

Geocodificação reversa em um clique

ReverseGeocode retorna os lugares a até 50 km de uma coordenada, primeiro os lugares povoados e cada grupo ordenado por distância, e ReverseGeocodeCountry retorna o código ISO do país do lugar mais próximo.

```
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 mantém uma conexão SQLite, portanto use um componente a partir de uma thread por vez; crie um por thread para consultas paralelas.

Disponibilidade

Framework	Versão mínima
VCL, FireMonkey, console e serviços	Delphi XE7 (FireDAC SQLite)
Lazarus	Não disponível

Veja Também

- TsgcMapGeoIndexBuilder
- TsgcMapOfflineGeocoder
- TsgcMapGeocoder

Grafos de Rota e Roteamento Offline

O roteamento sem conexão de rede precisa da malha viária em disco em uma forma que um roteador consiga pesquisar rapidamente. O sgcMaps faz isso em duas etapas:

1. **TsgcMapRouteGraphBuilder** (unit sgcMaps_OfflineRouter) lê dados do OpenStreetMap, um arquivo .osm.pbf ou .osm, e grava um **grafo de rotas** binário compacto, um arquivo .sgcgraph com os perfis de carro, bicicleta e a pé.
2. TsgcMapOfflineRouter carrega o grafo, ajusta cada ponto de passagem à via mais próxima do perfil selecionado e encontra a rota com uma busca A* bidirecional.

Construindo o grafo

Build(input file, output file) roda em três fases, informadas por **Phase**: 1 lê as vias, 2 lê as coordenadas dos nós que essas vias usam, e 3 grava o grafo. **OnProgress** informa os bytes lidos e o tamanho total da entrada enquanto ela é lida. Quando termina, WayCount, NodeCount, MissingNodes, VertexCount, EdgeCount, ShapeCount e NameCount descrevem o grafo. O grafo armazena, para cada via, seu comprimento, sua velocidade para carros, os sentidos em que pode ser percorrida para cada perfil, seus pontos intermediários e seu nome.

```
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;
```

Construa o grafo uma vez, para a região que a aplicação cobre, e distribua o arquivo .sgcgraph. A ferramenta de console RouteGraphBuilder faz o mesmo pela linha de comando.

Roteamento

Defina GraphFile e Profile (morCar, morBike ou morFoot) e chame Open. **GetRoute** recebe dois ou mais pontos de passagem e preenche um **TsgcMapRoute**, a classe usada pelo TsgcMapDirections online, com a geometria, a distância, a duração, um trecho para cada par de pontos de passagem e os passos de manobra. **sgcMapAddRouteToPolylines** a desenha em um 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;
```

O roteador offline também pode ser o motor de TsgcMapRouteCalculator, por meio de seu evento OnCalculate-Route, o que dá ao usuário pontos de passagem arrastáveis sem conexão de rede.

Disponibilidade

Framework	Versão mínima
VCL, FireMonkey, console e serviços	Delphi XE2
Lazarus	Não disponível

Veja Também

- Leitor de OpenStreetMap
- TsgcMapOfflineRouter
- TsgcMapRouteCalculator

TsgcMapService

Unidade: sgcMaps_Services
A classe base de todo serviço de mapa online: TsgcMapGeocoder, TsgcMapDirections, TsgcMapPlaces, TsgcMapElevation e TsgcMapTimeZone. Ela não é registrada na paleta. Os serviços enviam requisições HTTPS com THTTPClient e analisam as respostas com System.JSON, portanto exigem o Delphi XE8 ou posterior e não estão disponíveis no Lazarus.

Chamadas síncronas e assíncronas

- Os métodos **síncronos** (Geocode, GetRoute, Suggest...) bloqueiam a thread que os chama até a resposta chegar. Eles nunca geram exceção: em caso de falha retornam False e definem **LastError**.
- Os métodos **assíncronos** (GeocodeAsync, GetRouteAsync, SuggestAsync...) executam a requisição em uma thread em segundo plano e disparam o evento de resultado, ou **OnError**, na thread principal por meio de TThread.Queue. Aplicações VCL e FireMonkey não precisam de mais nada. Uma aplicação de console ou um serviço deve chamar CheckSynchronize periodicamente, ou usar os métodos síncronos.
- O componente deve permanecer vivo até que uma requisição assíncrona tenha disparado seu evento. Os objetos de resultado passados aos eventos pertencem ao componente e são liberados quando o evento retorna, portanto copie o que precisar manter.

Servidores públicos e políticas de uso

Os provedores padrão são serviços públicos gratuitos com políticas de uso. O Nominatim, o servidor de demonstração do OSRM e a API pública do OpenTopoData permitem no máximo uma requisição por segundo, e os componentes impõem esse intervalo. Eles também esperam um **UserAgent** identificador. Use seu próprio servidor ou um provedor comercial em produção e para requisições em massa.

Propriedades

Propriedade	Tipo	Padrão	Descrição
ApiKey	string		Chave dos provedores que precisam de uma. Para o provedor de fuso horário GeoNames, é o nome de usuário do GeoNames.
Language	string	en	Idioma dos resultados, para os provedores que o suportam.
Timeout	Integer	15000	Timeout da requisição em milissegundos.
UserAgent	string	sgcMaps/2.0	Cabeçalho User-Agent de cada requisição. Identifique sua aplicação aqui.
Referer	string		Cabeçalho Referer, enviado quando não está vazio.
LastError	string		Erro da última requisição que falhou. Somente leitura.

Eventos

Evento	Descrição
OnError	Uma requisição assíncrona falhou. Disparado na thread principal com o texto do erro.

Exemplo: Configuração comum dos serviços

Um procedimento configura todos os serviços de um formulário, já que as propriedades são declaradas na classe 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;
```

Veja Também

- TsgcMapGeocoder
- TsgcMapDirections
- TsgcMapPlaces
- TsgcMapElevation
- TsgcMapTimeZone

TsgcMapGeocoder

Unidade: sgcMaps_Geocoding
Transforma um endereço ou um nome de lugar em coordenadas (geocodificação) e coordenadas em um endereço (geocodificação reversa).
O componente descende de TsgcMapService e herda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** e **OnError**. Exige o Delphi XE8 ou posterior. Os métodos síncronos nunca geram exceção: retornam False e definem LastError. Os métodos assíncronos disparam seus eventos na thread principal.

Provedores

Provedor	Serviço	Chave
mgpNominatim	Nominatim (OpenStreetMap). O padrão. NominatimURL aponta para uma instância própria.	Não
mgpGoogle	Google Geocoding API	Sim
mgpHere	HERE Geocoding and Search	Sim
mgpAzure	Azure Maps Search	Sim
mgpMapBox	Mapbox Geocoding	Sim
mgpOpenRouteService	OpenRouteService Geocode	Sim
mgpGeoApify	Geoapify Geocoding	Sim

Propriedades

Propriedade	Tipo	Padrão	Descrição
Provider	TsgcMapGeocodingProvider	mgpNominatim	O serviço de geocodificação.
NominatimURL	string	https://nominatim.openstreetmap.org	URL base do servidor Nominatim.
MaxResults	Integer	5	Número máximo de resultados.
CountryCodes	string		Filtro de países separado por vírgulas com códigos ISO 3166-1 alfa-2 ('es,fr'). O HERE espera códigos alfa-3 ('ESP,FRA') e o Azure o ignora.

Métodos

Método	Descrição

Geocode(const aQuery: string; aResults: TsgcMapGeocodingResults): Boolean	Pesquisa um endereço ou um lugar. True quando pelo menos um resultado foi encontrado.
ReverseGeocode(aLat, aLon: Double; aResults: TsgcMapGeocodingResults): Boolean	Encontra o endereço em uma coordenada.
GeocodeAsync(const aQuery: string)	Geocode assíncrono. Os resultados chegam em OnGeocodeResult.
ReverseGeocodeAsync(aLat, aLon: Double)	ReverseGeocode assíncrono. Os resultados chegam em OnGeocodeResult.
BuildURL, BuildReverseURL	A URL da requisição para o provedor atual, útil para log e testes.

Eventos

Evento	Descrição
OnGeocodeResult	Os resultados de uma requisição assíncrona, na thread principal. A lista é liberada quando o evento retorna.

Resultados

Cada resultado é um TsgcMapGeocodingResult com **Latitude**, **Longitude**, **DisplayName**, **Street**, **HouseNumber**, **PostalCode**, **City**, **State**, **Country**, **CountryCode** (ISO alfa-2 quando o provedor o retorna, alfa-3 para o HERE), **Confidence** (0 a 1, 0 quando desconhecida) e a caixa delimitadora **BoundsNorth**, **BoundsSouth**, **BoundsEast** e **BoundsWest** (veja HasBounds). As mesmas classes de resultado são usadas por TsgcMapOfflineGeocoder.

Exemplo: Geocodificação e geocodificação reversa

```
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;
```

Veja Também

- TsgcMapService
- TsgcMapPlaces
- TsgcMapOfflineGeocoder

TsgcMapDirections

Unidade: sgcMaps_Directions
Calcula uma rota por dois ou mais pontos de passagem com um serviço de roteamento online. As distâncias são em metros e as durações em segundos.
O componente descende de TsgcMapService e herda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** e **OnError**. Exige o Delphi XE8 ou posterior. Os métodos síncronos nunca geram exceção: retornam False e definem LastError. Os métodos assíncronos disparam seus eventos na thread principal.

Provedores

Provedor	Serviço	Chave
mdpOSRM	OSRM. O OSRMURL padrão é o servidor de demonstração público do projeto OSRM, que permite apenas uso leve e atende somente o perfil de carro. Use seu próprio osrm-routed em produção e para caminhada e ciclismo.	Não
mdpOpenRouteService	OpenRouteService Directions	Sim
mdpGoogleRoutes	Google Routes API	Sim
mdpHere	HERE Routing	Sim
mdpAzure	Azure Maps Route	Sim
mdpMapBox	Mapbox Directions	Sim
mdpGeoApify	Geoapify Routing	Sim

Todo provedor retorna as instruções passo a passo em **Language**, exceto o OSRM, que não retorna nenhuma: para o OSRM o componente constrói instruções curtas em inglês.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Provider	TsgcMapDirectionsProvider	mdpOSRM	O serviço de roteamento.
OSRMURL	string	https://router.project-osrm.org	URL base do servidor OSRM.
TravelMode	TsgcMapTravelMode	mtmDriving	mtmDriving, mtmWalking, mtmCycling ou mtmTruck. IsTravelModeSupported informa quais modos um provedor aceita.
Alternatives	Boolean	False	Solicita rotas alternativas.

AvoidTolls, AvoidHighways, AvoidFerries	Boolean	False	Restrições da rota, para os provedores que as suportam.
OptimizeWaypoints	Boolean	False	Permite que o provedor reordene os pontos de passagem intermediários.

Métodos

Método	Descrição
GetRoute(const aWaypoints: array of TsgcMapCoord; aRoutes: TsgcMapRoutes): Boolean	Calcula a rota. True quando pelo menos uma rota foi encontrada; o índice 0 é a rota principal.
GetRouteAsync(const aWaypoints: array of TsgcMapCoord)	GetRoute assíncrono. As rotas chegam em OnRouteResult.
ValidateRequest(const aWaypoints: array of TsgcMapCoord): string	Uma string vazia quando a requisição pode ser enviada, caso contrário o motivo pelo qual não pode.
IsTravelModeSupported(aProvider; aMode): Boolean	Função de classe: se um provedor suporta um modo de viagem.

Eventos

Evento	Descrição
OnRouteResult	As rotas de uma requisição assíncrona, na thread principal. A lista é liberada quando o evento retorna.

Resultados de rota

Classe	Membros
TsgcMapRoutes	Count e Items[Index] (propriedade padrão) de TsgcMapRoute.
TsgcMapRoute	Distance, Duration, Summary, a geometria (Points, PointCount, GetPoints), os trechos (Legs, LegCount) e a caixa delimitadora (BoundsNorth, BoundsSouth, BoundsEast, BoundsWest, CalcBounds).
TsgcMapRouteLeg	A parte da rota entre dois pontos de passagem consecutivos: Distance, Duration, Summary, Steps e StepCount.
TsgcMapRouteStep	Uma manobra: Instruction, Maneuver, StreetName, Distance, Duration, Latitude, Longitude e PointIndex (sua posição na geometria da rota).

sgcMapAddRouteToPolylines(aRoute, aPolylines, aColor, aWidth) adiciona a um mapa uma polilinha com a geometria de uma rota e a retorna.

Exemplo: Calculando uma rota

```
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;
```

Veja Também

- TsgcMapService
- TsgcMapRouteCalculator
- TsgcMapOfflineRouter

TsgcMapPlaces

Unidade: sgcMaps_Places
Preenchimento automático de lugares: sugestões para o texto que o usuário está digitando, normalmente exibidas em uma lista sob uma caixa de busca.
O componente descende de TsgcMapService e herda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** e **OnError**. Exige o Delphi XE8 ou posterior. Os métodos síncronos nunca geram exceção: retornam False e definem LastError. Os métodos assíncronos disparam seus eventos na thread principal.

Provedores

Provedor	Serviço	Chave
mplPhoton	Photon (komoot). O padrão. Aceita apenas os idiomas de, en e fr; qualquer outro Language não é enviado.	Não
mplNominatim	Nominatim. Limitado a uma requisição por segundo, portanto não use o servidor público para autocompletar a cada tecla.	Não
mplGoogle	Google Places API (New). As sugestões não têm coordenadas: chame Retrieve.	Sim
mplHere	HERE Autosuggest	Sim
mplMapBox	Mapbox Search Box API. As sugestões não têm coordenadas: chame Retrieve.	Sim
mplGeoApify	Geoapify Autocomplete	Sim

Propriedades

Propriedade	Tipo	Padrão	Descrição
Provider	TsgcMapPlacesProvider	mplPhoton	O serviço de preenchimento automático.
NominatimURL	string	https://nominatim.openstreetmap.org	URL base do servidor Nominatim.
MaxResults	Integer	5	Número máximo de sugestões.
BiasLatitude, BiasLongitude	Double	0	Resultados próximos a esta posição são preferidos e têm seu Distance preenchido. Não usado quando ambos são 0.
BiasRadius	Double	0	Raio em torno da posição de viés, em metros. 0 significa sem raio. Usado por Google, HERE e Nominatim.
SessionToken	string		Token da sessão de busca atual. Público.

Métodos

Método	Descrição
Suggest(const aText: string; aResults: TsgcMapPlaceResults): Boolean	Retorna as sugestões para um texto. True quando pelo menos uma foi encontrada.
SuggestAsync(const aText: string)	Suggest assíncrono. As sugestões chegam em OnSuggestResult.
Retrieve(aPlace: TsgcMapPlace): Boolean	Preenche as coordenadas, e um nome ou endereço ausente, de uma sugestão. Apenas o Google e o Mapbox fazem uma requisição; para os outros provedores retorna HasCoordinates.
NewSession	Inicia uma nova sessão de busca com um novo SessionToken.

Eventos

Evento	Descrição
OnSuggestResult	As sugestões de uma requisição assíncrona, na thread principal. A lista é liberada quando o evento retorna.

Resultados

Cada sugestão é um TsgcMapPlace com **Id** (o identificador do provedor, usado por Retrieve), **Name**, **Address**, **Latitude**, **Longitude**, **HasCoordinates**, **Category** e **Distance** (metros a partir da posição de viés, -1 quando desconhecida).

Exemplo: Sugestões enquanto se digita

```
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;
```

Veja Também

- TsgcMapService
- TsgcMapGeocoder

TsgcMapElevation

Unidade: sgcMaps_Elevation
Retorna a elevação do terreno de uma lista de pontos, ou de amostras igualmente espaçadas ao longo de um caminho, por exemplo um perfil de elevação de uma rota. Listas longas são divididas em lotes automaticamente.
O componente descende de TsgcMapService e herda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** e **OnError**. Exige o Delphi XE8 ou posterior. Os métodos síncronos nunca geram exceção: retornam False e definem LastError. Os métodos assíncronos disparam seus eventos na thread principal.

Provedores

Provedor	Serviço	Pontos por requisição	Chave
meOpenElevation	Open-Elevation	200	Não
meOpenTopoData	OpenTopoData. O padrão. A API pública é limitada a uma requisição por segundo.	100	Não
meGoogle	Google Elevation API	256	Sim

Propriedades

Propriedade	Tipo	Padrão	Descrição
Provider	TsgcMapElevationProvider	meOpenTopoData	O serviço de elevação.
Dataset	string	srtm90m	Conjunto de dados do OpenTopoData: srtm90m, srtm30m, aster30m, eudem25m e outros.

Métodos

Método	Descrição
GetElevations(const aPoints: array of TsgcMapCoord; aResults: TsgcMapElevationResults): Boolean	Um resultado por ponto, na mesma ordem.
GetElevationAlongPath(const aPoints: array of TsgcMapCoord; aSamples: Integer; aResults: TsgcMapElevationResults): Boolean	aSamples pontos (pelo menos 2, incluindo o primeiro e o último) igualmente espaçados por distância ao longo da polilinha aPoints.
GetElevationsAsync, GetElevationAlongPathAsync	Versões assíncronas. Os resultados chegam em OnElevationResult.
BatchSize: Integer	Máximo de pontos por requisição do provedor atual.
SamplePath	Função de classe: as amostras igualmente espaçadas de uma polilinha e suas distâncias a partir do primeiro ponto.

Eventos

Evento	Descrição
OnElevationResult	Os resultados de uma requisição assíncrona, na thread principal. A lista é liberada quando o evento retorna.

Resultados

Cada resultado é um TsgcMapElevationResult com **Latitude**, **Longitude**, **Elevation** (metros acima do nível do mar), **HasElevation** (False quando o provedor não tem dados para o ponto) e **Distance** (metros a partir do primeiro ponto, preenchido por GetElevationAlongPath).

Exemplo: Perfil de elevação

```
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;
```

Veja Também

- TsgcMapService
- TsgcMapDirections

TsgcMapTimeZone

Unidade: sgcMaps_TimeZone
Retorna o fuso horário de uma coordenada: seu identificador IANA, o deslocamento em relação ao UTC e a hora local.
O componente descende de TsgcMapService e herda **ApiKey**, **Language**, **Timeout**, **UserAgent**, **Referer**, **LastError** e **OnError**. Exige o Delphi XE8 ou posterior. Os métodos síncronos nunca geram exceção: retornam False e definem LastError. Os métodos assíncronos disparam seus eventos na thread principal.

Provedores

Provedor	Serviço	Chave
mtzTimeAPI	TimeAPI (timeapi.io). O padrão.	Não
mtzGeoNames	GeoNames. ApiKey é o nome de usuário do GeoNames.	Nome de usuário
mtzGoogle	Google Time Zone API	Sim
mtzAzure	Azure Maps Timezone	Sim

O parâmetro opcional **aUtcTime** (0 significa agora) é respeitado pelo Google e pelo Azure. TimeAPI e GeoNames sempre respondem para a hora atual.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Provider	TsgcMapTimeZoneProvider	mtzTimeAPI	O serviço de fuso horário.

Métodos

Método	Descrição
GetTimeZone(aLat, aLon: Double; aInfo: TsgcMapTimeZoneInfo; aUtcTime: TDateTime = 0): Boolean	Preenche aInfo com o fuso horário da coordenada.
GetTimeZoneAsync(aLat, aLon: Double; aUtcTime: TDateTime = 0)	GetTimeZone assíncrono. A resposta chega em OnTimeZoneResult.
UtcNow: TDateTime	Função de classe: a hora UTC atual.

Eventos

Evento	Descrição
OnTimeZoneResult	A resposta de uma requisição assíncrona, na thread principal. O objeto é liberado quando o evento retorna.

Resultados

TsgcMapTimeZoneInfo tem **TimeZoneId** ('Europe/Madrid'), **Name** (o nome de exibição quando o provedor retorna um, caso contrário o TimeZoneId), **UtcOffsetSeconds** (padrão mais horário de verão), **DstOffsetSeconds** (a parte do horário de verão), **IsDst** e **LocalTime** (ISO 8601 sem deslocamento).

Exemplo: Fuso horário de uma 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;
```

Veja Também

- TsgcMapService

TsgcMapRouteCalculator

Unidade: sgcMaps_RouteCalculator
Edição interativa de rotas em um mapa. Os pontos de passagem são marcadores arrastáveis em **Markers** e a rota é uma polilinha em **Polylines**; aponte ambos para as coleções de um controle de mapa. Arrastar um ponto de passagem, adicionar um ou remover um recalcula a rota quando **AutoCalculate** é True. Exige o Delphi XE2 ou posterior; a propriedade Directions exige o Delphi XE8.
O motor de rotas é o evento **OnCalculateRoute**, em que qualquer motor pode preencher a rota (por exemplo TsgcMapOfflineRouter), ou, quando o evento não a trata, o TsgcMapDirections online atribuído a **Directions**.
Os marcadores de ponto de passagem recebem Tag = MarkerTagBase + índice e a polilinha da rota recebe Tag = MarkerTagBase. Com ShowAlternatives as rotas alternativas são desenhadas como polilinhas cinzas mais finas com Tag = MarkerTagBase + 1.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Markers	TsgcMapMarkers		Coleção dos marcadores de ponto de passagem, por exemplo Map.Markers.
Polylines	TsgcMapPolylines		Coleção das polilinhas de rota, por exemplo Map.Polylines.
Directions	TsgcMapDirections		Motor de rotas online, usado quando OnCalculateRoute não trata a requisição.
Active	Boolean	True	False desvincula o calculador de Markers e Polylines: sem acompanhar arrastes de marcadores e sem cálculo automático.
AutoCalculate	Boolean	True	Recalcula após cada mudança dos pontos de passagem.
ShowAlternatives	Boolean	False	Desenha também as rotas alternativas.
RouteColor, RouteWidth	TColor, Integer	blue, 4	Cor e largura da polilinha da rota.
WaypointColor	TColor	red	Cor dos marcadores de ponto de passagem.
MarkerTagBase	Integer	SGC_MAP_ROUTE_TAG_BASE	Primeiro Tag usado para os marcadores e polilinhas do calculador.
Waypoints[Index], WaypointMarkers[Index]	TsgcMapCoord, TsgcMapMarker		Os pontos de passagem e seus marcadores. Somente leitura.

Route, RoutePolyline, Routes	TsgcMapRoute, TsgcMapPolyline, TsgcMapRoutes		A rota principal, sua polilinha e todas as rotas do último cálculo (o índice 0 é a rota principal). Somente leitura.

Métodos

Método	Descrição
AddWaypoint(aLat, aLon: Double): Integer	Acrescenta um ponto de passagem e retorna seu índice.
InsertWaypoint(aIndex: Integer; aLat, aLon: Double)	Insere um ponto de passagem.
AddWaypointNearestSegment(aLat, aLon: Double): Integer	Insere um ponto de passagem entre os dois pontos de passagem cujo trecho está mais próximo do ponto, como faz o arraste da linha da rota.
MoveWaypoint(aIndex: Integer; aLat, aLon: Double)	Move um ponto de passagem.
RemoveWaypoint(aIndex: Integer), ClearWaypoints	Remove um ou todos os pontos de passagem.
WaypointCount: Integer	Número de pontos de passagem.
Calculate: Boolean	Cálculo síncrono: OnCalculateRoute, depois Directions.GetRoute.
CalculateAsync	OnCalculateRoute, aplicado imediatamente quando tratado, caso contrário Directions.GetRouteAsync. Os eventos disparam na thread principal.
SelectRoute(aIndex: Integer)	Torna uma rota alternativa a rota principal, redesenha e dispara OnRouteCalculated.

Eventos

Evento	Descrição
OnCalculateRoute	Disparado primeiro em todo cálculo. Preencha aRoute e defina aHandled como True para usar seu próprio motor.
OnRouteCalculated	Uma nova rota foi calculada e desenhada.
OnWaypointsChanged	Os pontos de passagem mudaram.
OnError	O cálculo falhou.

Exemplo: Construindo uma rota clicando no 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 a rota sem conexão de rede, trate OnCalculateRoute com o roteador offline, como mostrado em TsgcMapOfflineRouter.

Veja Também

- TsgcMapDirections
- TsgcMapOfflineRouter
- TsgcVCLMap

TsgcMapOfflineRouter

Unidade: sgcMaps_OfflineRouter
Calcula rotas sem conexão de rede, em um arquivo de grafo de rotas (**.sgcgraph**) construído a partir de dados do OpenStreetMap. Exige o Delphi XE2 ou posterior.
Construa o grafo uma vez a partir de um extrato do OpenStreetMap (.osm.pbf, ou XML .osm) com **TsgcMapRouteGraphBuilder**, declarado na mesma unit, ou com a ferramenta RouteGraphBuilder em Tools\Maps. O roteador preenche a mesma classe TsgcMapRoute que TsgcMapDirections, de modo que uma rota de qualquer um dos motores é desenhada e inspecionada da mesma forma. Uma rota por vários pontos de passagem tem um trecho para cada par de pontos de passagem consecutivos.

Propriedades

Propriedade	Tipo	Padrão	Descrição
GraphFile	string		O arquivo .sgcgraph.
Profile	TsgcMapOfflineRouterProfile	morCar	morCar, morBike ou morFoot.
Active	Boolean	False	Abre ou fecha o grafo.
LastError	string		Erro da última chamada que falhou. Somente leitura.
VertexCount, EdgeCount	Integer		Tamanho do grafo carregado. Somente leitura.

Métodos

Método	Descrição
Open, Close	Carrega ou libera o grafo.
GetRoute(const aWaypoints: array of TsgcMapCoord; aRoute: TsgcMapRoute): Boolean	Calcula a rota. False quando nenhuma rota foi encontrada, com o motivo em LastError.

Exemplo: Rota offline e o motor offline do calculador de rotas

```
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;
```

Veja Também

- TsgcMapRouteCalculator
- TsgcMapDirections

TsgcMapOfflineGeocoder

Unidade: sgcMaps_OfflineGeocoder
Geocodificação e geocodificação reversa sem conexão de rede, em um índice SQLite construído a partir dos dumps do GeoNames por TsgcMapGeoIndexBuilder. Usa o driver FireDAC SQLite, portanto exige o Delphi XE7 ou posterior.
As buscas diretas aceitam um nome de lugar, um prefixo e um país ou região opcional: 'Madrid', 'mostol', 'Toledo, US', 'Madrid ES' ou 'Toledo, Ohio'. Os resultados são ordenados por correspondência exata antes de correspondência de prefixo, depois por população, depois por código de feição (capital, depois sede administrativa, depois outros lugares povoados). As buscas reversas retornam os lugares mais próximos a até 50 km, primeiro os lugares povoados, cada grupo ordenado por distância.
Cada componente possui uma conexão SQLite e deve ser usado a partir de uma thread por vez. Para consultar o mesmo índice a partir de várias threads, crie um componente por thread; o arquivo é aberto somente para leitura.

Propriedades

Propriedade	Tipo	Padrão	Descrição
DatabaseFile	string		O arquivo de índice SQLite.
Active	Boolean	False	Abre ou fecha o índice.
MaxResults	Integer	5	Número máximo de resultados.
CountryCodes	string		Códigos ISO alfa-2 separados por vírgulas que restringem toda busca.
Language	string		Dica de idioma. Os dumps do GeoNames não trazem idioma por nome alternativo, portanto o valor é atualmente ignorado.

Métodos

Método	Descrição
Open, Close	Abre ou fecha o índice.
Geocode(const aQuery: string; aResults: TsgcMapGeocodingResults): Boolean	Busca direta. True quando pelo menos um lugar foi encontrado.
ReverseGeocode(aLatitude, aLongitude: Double; aResults: TsgcMapGeocodingResults): Boolean	Os lugares mais próximos a até 50 km.
ReverseGeocodeCountry(aLatitude, aLongitude: Double): string	Código ISO do país do lugar mais próximo, vazio quando não há nenhum a até 50 km.

Os resultados usam a mesma classe TsgcMapGeocodingResult que TsgcMapGeocoder.

Exemplo: Busca offline

```
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;
```

Veja Também

- TsgcMapGeoIndexBuilder
- TsgcMapGeocoder

TsgcMapGeoIndexBuilder

Unidade: sgcMaps_OfflineGeocoder
Constrói o índice SQLite que TsgcMapOfflineGeocoder pesquisa, a partir de um dump do GeoNames baixado de https://download.geonames.org/export/dump/: cities500.zip, cities1000.zip, allCountries.zip, um arquivo de país como ES.zip, ou o arquivo .txt extraído. Exige o Delphi XE7 ou posterior (FireDAC SQLite).
A construção roda na thread que chama. A ferramenta GeoIndexBuilder em Tools\Maps encapsula este componente em uma aplicação de console.

Propriedades

Propriedade	Tipo	Padrão	Descrição
InputFile	string		O dump do GeoNames: o arquivo .txt separado por tabulações ou um .zip que o contém.
DatabaseFile	string		O arquivo SQLite a criar. Um arquivo existente é sobrescrito.
Admin1CodesFile	string		admin1CodesASCII.txt opcional, para os nomes das regiões de primeiro nível.
CountryInfoFile	string		countryInfo.txt opcional, para os nomes dos países.
FeatureClasses	string	PA	Feature classes do GeoNames a importar: P lugares povoados, A áreas administrativas, S pontos, T terreno, L áreas, H hidrografia, R estradas, U submarino, V vegetação.
MinPopulation	Int64	0	Lugares povoados abaixo dessa população são ignorados. As outras classes não são filtradas.
IncludeAlternateNames	Boolean	True	Indexa os nomes alternativos de até 64 caracteres, como nomes curtos e nomes em outros idiomas.
BatchSize	Integer	2000	Linhas gravadas por lote.
RowsRead, PlacesWritten, NamesWritten, ElapsedMs	Int64		Estatísticas da última construção. Somente leitura.

Métodos

Método	Descrição
Build	Constrói o índice.
RowsPerSecond: Double	Velocidade da última construção.

Eventos

Evento	Descrição
OnProgress	Disparado durante a construção com as linhas lidas, os lugares gravados e os milissegundos decorridos.

Exemplo: Construindo um 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;
```

Veja Também

- TsgcMapOfflineGeocoder

TsgcMapLocation

Unidade: sgcMaps_Location
A posição do dispositivo. Ela encapsula o TLocationSensor de System.Sensors, portanto exige o Delphi XE6 ou posterior e funciona em aplicações VCL e FireMonkey: o Windows usa a Windows Location API, e o Android, o iOS e o macOS usam seus serviços de localização nativos.
O componente não conhece nenhum mapa. **OnLocationChanged** informa cada leitura, e **UpdateMarker** e **UpdateAccuracyCircle** mantêm um marcador de posição e um círculo de precisão nas coleções de um mapa. Com **FollowMode** definido como mfmCenter, cada leitura também dispara **OnFollow**, onde a aplicação centraliza seu mapa.

Permissões

- Android: declare ACCESS_FINE_LOCATION em Uses Permissions nas opções do projeto e solicite-a em tempo de execução antes de definir Active como True, com **RequestPermission** (Delphi 10.3 ou posterior) ou com PermissionsService.RequestPermissions.
- iOS e macOS: adicione as chaves de descrição de uso de localização às informações de versão do projeto.

Propriedades

Propriedade	Tipo	Padrão	Descrição
Active	Boolean	False	Inicia ou para o sensor de localização.
Accuracy	Double		Precisão desejada do sensor, em metros.
Distance	Double		Movimento mínimo, em metros, entre duas leituras informadas.
FollowMode	TsgcMapLocationFollowMode	mfmNone	mfmCenter dispara OnFollow após cada leitura.
Available	Boolean		Se o dispositivo tem um sensor de localização. Somente leitura.
HasLocation	Boolean		Se uma leitura já chegou. Somente leitura.
LastLatitude, LastLongitude, LastAccuracy, LastHeading, LastSpeed, LastTimestamp			A última leitura. Somente leitura.
TrackCount, TrackIndex, TrackIntervalMs, TrackAccuracy			Estado do trajeto simulado carregado com LoadTrack. Somente leitura.

Métodos

Método	Descrição
UpdateMarker(aMarkers: TsgcMapMarkers; aTag: Integer)	Adiciona ou move o marcador de posição, identificado por seu Tag.

UpdateAccuracyCircle(aCircles: TsgcMapCircles; aTag: Integer)	Adiciona ou atualiza um círculo com o raio da precisão.
RequestPermission	Solicita a permissão de localização no Android. A resposta chega em OnPermissionResult.
SimulateLocation(aLatitude, aLongitude: Double; aAccuracy: Double = 0; aHeading: Double = -1; aSpeed: Double = -1)	Informa uma posição como se o sensor a tivesse lido, para testes.
LoadTrack(const aPoints: array of TsgcMapCoord; aIntervalMs: Integer = 1000; aAccuracy: Double = 10)	Carrega uma lista de pontos, por exemplo os pontos de uma rota, para simular uma viagem.
StepTrack: Boolean	Informa o próximo ponto do trajeto carregado. False no fim do trajeto.
ResetTrack	Volta ao primeiro ponto do trajeto.

Eventos

Evento	Descrição
OnLocationChanged	Uma nova leitura: latitude, longitude, precisão, direção e velocidade.
OnFollow	Com FollowMode = mfmCenter, disparado após OnLocationChanged com a posição em que centralizar.
OnStateChanged	O sensor foi iniciado ou parado.
OnError	O sensor informou um erro.
OnPermissionResult	A resposta a RequestPermission.

Exemplo: Mostrando e acompanhando a posição

```
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;
```

Veja Também

- TsgcVCLMap
- TsgcFMXMap

Ferramentas

A pasta **Tools\Maps** tem dois aplicativos de console que preparam os dados dos componentes offline. Eles encapsulam as classes construtoras do sgcMaps, de modo que o mesmo trabalho pode ser feito a partir do seu próprio código.

Ferramenta	Entrada	Saída	Usado por	Versão mínima
GeoIndexBuilder	Um dump do GeoNames	Um índice SQLite	TsgcMapOfflineGeocoder	Delphi XE7 (FireDAC SQLite)
RouteGraphBuilder	Um extrato do OpenStreetMap	Um arquivo .sgcgraph	TsgcMapOfflineRouter	Delphi XE2

Compilando as ferramentas

Cada ferramenta é um único arquivo .dpr. Abra-o no IDE e compile, ou compile-o em um prompt de comando com a pasta Source nos caminhos de units e de include, por exemplo:

```
dcc32 -NSSystem;Data;FireDAC -U..\..\..\Source -I..\..\..\Source sgcMapsGeoIndexBuilder.dpr
```

Construa os dados uma vez, em uma máquina de desenvolvimento, e distribua os arquivos resultantes com sua aplicação.

GeoIndexBuilder

sgcMapsGeoIndexBuilder (Tools\Maps\GeoIndexBuilder) constrói o índice SQLite do geocodificador offline a partir de um dump do GeoNames. É um encapsulamento de console de TsgcMapGeoIndexBuilder e exige o Delphi XE7 ou posterior, pois grava o índice com o FireDAC SQLite.

Dados de entrada

Os dumps do GeoNames são downloads gratuitos em https://download.geonames.org/export/dump/:

- **cities500.zip**, **cities1000.zip**, **cities5000.zip** ou **cities15000.zip**: os lugares povoados do mundo acima dessa população.
- **XX.zip**, como ES.zip: todas as feições de um país.
- **allCountries.zip**: todas as feições do mundo.
- **admin1CodesASCII.txt** (opcional): os nomes das áreas administrativas de primeiro nível, como estados e regiões.
- **countryInfo.txt** (opcional): os nomes dos países, usados para qualificar buscas como "Paris, France".

A ferramenta lê diretamente o arquivo .zip, ou o arquivo .txt extraído.

Linha de comando

```
sgcMapsGeoIndexBuilder <input.zip|txt> <output.sqlite> [--minpop N]
  [--admin1 file] [--countryinfo file] [--classes PA] [--noalternates]
```

Parâmetro	Propriedade	Significado
input	InputFile	O dump do GeoNames, .zip ou .txt.
output	DatabaseFile	O arquivo SQLite a criar. Um arquivo existente é sobrescrito.
--minpop N	MinPopulation	Ignora os lugares povoados (feature class P) abaixo dessa população. As outras feature classes não são filtradas.
--admin1 file	Admin1CodesFile	O arquivo admin1CodesASCII.txt.
--countryinfo file	CountryInfoFile	O arquivo countryInfo.txt.
--classes	FeatureClasses	As feature classes do GeoNames a importar. O padrão é PA: lugares povoados e áreas administrativas. As outras são S (pontos), T (terreno), L (áreas), H (hidrografia), R (estradas), U (submarino) e V (vegetação).
--noalternates	IncludeAlternateNames	Omite os nomes alternativos (nomes curtos e nomes em outros idiomas), o que deixa o índice menor.

```
sgcMapsGeoIndexBuilder cities500.zip world.sqlite --admin1 admin1CodesASCII.txt --countryinfo countryInfo.txt
```

Saída

A ferramenta imprime seu progresso, depois as linhas lidas, os lugares e nomes gravados, o tempo e o tamanho do índice:

```
Building world.sqlite from cities500.zip
  ... rows read, ... places, ... s
Done: ... rows read, ... places, ... names, ... s, ... rows/s, ... MB
```

Termina com o código 0 em caso de sucesso, 1 para parâmetros errados e 2 quando a construção falha, depois de imprimir o erro.

Usando o índice

Distribua o arquivo .sqlite com a aplicação e abra-o com TsgcMapOfflineGeocoder definindo seu DatabaseFile. O componente passa então a responder a Geocode, ReverseGeocode e ReverseGeocodeCountry sem conexão de rede. A aplicação precisa do driver FireDAC SQLite em tempo de execução, como descrito em Solução de Problemas. Índice de Geocodificação Offline explica o conteúdo do índice.

RouteGraphBuilder

sgcMapsRouteGraphBuilder (Tools\Maps\RouteGraphBuilder) transforma um extrato do OpenStreetMap no grafo de rotas do roteador offline. É um encapsulamento de console de TsgcMapRouteGraphBuilder (unit sgcMaps_OfflineRouter) e exige o Delphi XE2 ou posterior.

Dados de entrada

Um extrato do OpenStreetMap no formato **.osm.pbf** ou como XML **.osm**. Serviços de download como o Geofabrik publicam extratos de todos os continentes, países e regiões. Escolha o menor extrato que cobre a área da sua aplicação, pois o tempo e a memória da construção crescem com o extrato.

Linha de comando

```
sgcMapsRouteGraphBuilder <input.osm.pbf|input.osm> <output.sgcgraph>

sgcMapsRouteGraphBuilder andorra-latest.osm.pbf andorra.sgcgraph
```

Saída

O construtor lê o arquivo duas vezes, primeiro as vias e depois as coordenadas dos nós que essas vias usam, e imprime seu progresso a cada 10 por cento. No final imprime as estatísticas do 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 conta os nós que as vias roteáveis referenciam, e missing conta os que não estão no arquivo, o que acontece com vias que cruzam a borda de um extrato. A ferramenta termina com o código 0 em caso de sucesso, 2 para parâmetros errados e 1 quando a construção falha, depois de imprimir o erro.

Usando o grafo

Um único arquivo .sgcgraph contém os perfis de carro, bicicleta e a pé. Distribua-o com a aplicação, defina-o como o GraphFile de TsgcMapOfflineRouter, escolha o Profile (morCar, morBike ou morFoot) e chame GetRoute. A rota volta na mesma classe TsgcMapRoute que TsgcMapDirections usa, de modo que é desenhada com sgcMapAddRouteToPolylines ou entregue ao TsgcMapRouteCalculator. Grafos de Rota e Roteamento Offline descreve o grafo e a classe construtora.

Lazarus

O sgcMaps tem um pacote para Lazarus e Free Pascal: **Packages\Lazarus\sgcMapsLazarus.lpk**. Ele instala o controle de mapa na LCL, com o mesmo modelo de mapa e o mesmo código do controle VCL, mas com menos recursos, porque as partes que precisam de units da RTL exclusivas do Delphi não são compiladas no FPC.
A diferença mais importante: **o controle Lazarus não consegue baixar tiles, portanto exibe apenas tiles de uma pasta.**

Instalando o pacote

1. Inicie o Lazarus e escolha **Package**, **Open Package File (.lpk)**.
2. Abra **Packages\Lazarus\sgcMapsLazarus.lpk** da instalação do sgcUI. O pacote encontra as units na pasta Source por meio de um caminho relativo, portanto mantenha a estrutura de pastas da instalação.
3. Pressione **Compile**, depois **Use**, **Install**, e deixe o Lazarus recompilar e reiniciar o IDE.
4. TsgcVCLMap agora está na página **SGC Maps** da paleta de componentes.

O pacote exige os pacotes LCL, FCL e IDEIntf, que toda instalação do Lazarus tem. Um projeto que usa o mapa lista sgcMapsLazarus em seus pacotes requeridos, o que o IDE adiciona quando você solta o controle em um formulário.

O que está incluído

Incluído	Unit
O controle de mapa na LCL, TsgcVCLMap	sgcMaps_VCL
O modelo de mapa: camadas, marcadores, polilinhas, polígonos, retângulos, círculos, rótulos, mapas de calor e opções	sgcMaps_Classes
A fonte de tiles de pasta (arquivos z/x/y)	sgcMaps_TileSource
A lista de provedores e sgcMapExpandTileURL	sgcMaps_Providers
As funções de geometria	sgcMaps_Geometry
O rasterizador de software, TsgcMapRasterizer	sgcMaps_Raster

O controle desenha marcadores (com ícones PNG personalizados), clusters, formas, rótulos e mapas de calor, e salva a visualização com RenderToBitmap, SaveToBMP e SaveToPNG.

O que não está incluído

- **Tiles de um servidor.** Uma camada com fonte mtsHTTP ou com um Provider não exibe tiles no Lazarus. Prepare uma pasta de tiles no Delphi com TsgcMapTileDownloader, ou com qualquer ferramenta que grave tiles z/x/y, e distribua-a com a aplicação.
- **MBTiles e tiles vetoriais.** Essas fontes precisam de FireDAC e System.ZLib.
- **GeoJSON, GPX e KML.** As units constam do pacote, mas compilam vazias no FPC.
- **Os componentes não visuais.** Os serviços online, o calculador de rotas, o roteador offline, o geocodificador offline, o leitor OSM e o TsgcMapLocation estão disponíveis somente no Delphi.
- **Rotação do mapa.** Options.Bearing é ignorado na LCL.
- **Opacidade.** A opacidade de camadas, polilinhas e preenchimentos não é aplicada na LCL.

Exemplo

O programa abaixo mostra um mapa a partir de uma pasta de tiles ao lado do executável. Crie um novo projeto Application, adicione sgcMapsLazarus a seus pacotes requeridos e substitua o 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.
```

A pasta **tiles** contém uma subpasta por nível de zoom, uma subpasta por coluna e uma imagem por tile, como tiles\12\2072\1529.png. Defina MaxZoom como o maior nível de zoom da pasta, para que o controle não procure tiles que não existem. Mantenha a atribuição dos tiles, pois os termos do provedor de tiles continuam valendo para uma cópia em disco.

Veja Também

- TsgcVCLMap
- Tiles Offline e o Baixador de Tiles
- Instalação

Visão Geral do .NET

sgcMaps for .NET é uma porta gerenciada em C# da biblioteca sgcMaps do Delphi, não um wrapper em torno dela. Ela tem o mesmo modelo de mapa, os mesmos provedores de tiles, os mesmos serviços online e o mesmo roteador e geocodificador offline, e desenha o mapa nativamente com SkiaSharp, sem controle de navegador e sem JavaScript.
Os arquivos são compatíveis entre as duas bibliotecas: um grafo de rota (.sgcgraph), um índice GeoNames ou uma pasta de tiles criados com o Delphi abrem no .NET, e vice-versa. A suíte de QA do .NET verifica isso com um grafo de rota e um índice GeoNames gravados pela biblioteca Delphi.

Pacotes

A porta é dividida em quatro pacotes NuGet, de modo que um projeto traz apenas o que usa. Os quatro têm a versão **2.0.0**: o sgcMaps for .NET mantém seus próprios números de versão, independentes da versão Delphi.

PackageId	Contém	Target Frameworks	Depende de
esegece.sgcMaps	O núcleo: o modelo de mapa (camadas, marcadores, polilinhas, polígonos, retângulos, círculos, rótulos, mapas de calor e opções), a matemática de tiles e a geometria, os provedores de tiles, as fontes de tiles de pasta, MBTiles e HTTP, o baixador de tiles, a importação e exportação de GeoJSON, GPX e KML, os leitores de tiles vetoriais e de estilo, e os serviços online: Geocoder, Di-	net8.0, net9.0	Microsoft.Data.Sqlite

	rections, Places, Elevation, MapTime-Zone e RouteCal-culator.		
esegece.sgcMaps.Rendering	**MapView**, o motor de mapa. Ele desenha ti-les raster e vetoriais, sobreposi-ções, clus-ters, ma-pas de ca-lor e po-pups com SkiaSharp, trata a en-trada de ponteiro, roda do mouse e teclado, e renderiza para uma imagem ou um arquivo PNG sem janela.	net8.0, net9.0	esegece.sgcMaps, SkiaSharp, SkiaSharp.NativeAssets.Linux.NoDependencies
esegece.sgcMaps.WinForms	**SgcMap-Control**, o controle de mapa Win-dows Forms, que hospeda um MapVi-ew.	net8.0-windows, net9.0-windows	esegece.sgcMaps.Rendering
esegece.sgcMaps.Offline	Os servi-ços offline: **OsmRea-der** (.osm.pbf e XML .osm), **Route-Graph-Builder** e **Offline-Router**, **GeoIndex-Builder** e **OfflineGe-ocoder**.	net8.0, net9.0	esegece.sgcMaps

Somente a edição Registered

O sgcMaps for .NET é um produto Registered (comercial). Não há edição Community nem pacote **.Community**: os quatro pacotes acima são os únicos.

Hosts

- **Windows Forms**: SgcMapControl, o equivalente do TsgcVCLMap do Delphi.
- **Sem janela**: o MapView renderiza em um canvas do SkiaSharp, em uma imagem ou em um arquivo PNG, no Windows, no Linux e no macOS. Isso cobre aplicações de console, serviços e servidores web.
- **Seu próprio host**: o MapView não depende de um framework de interface de usuário. Ele desenha em qualquer SKCanvas e recebe a entrada por PointerDown, PointerMove, PointerUp, Wheel, Pinch, Rotate e KeyDown, exatamente como o SgcMapControl o controla.

Não há controle WPF, Avalonia ou MAUI, nem equivalente do controle FireMonkey TsgcFMXMap.

Namespaces

Namespace	Conteúdo
esegece.sgcMaps	O modelo de mapa, MapCoord, MapColors, TileMath, MapGeometry, MapProviders, as fontes de tiles, TileDownloader, GeoJsonFormat, GpxFormat e KmlFormat.
esegece.sgcMaps.Services	Os serviços online (MapServiceBase, Geocoder, Directions, Places, Elevation, MapTimeZone), RouteCalculator e as classes de resultado, como GeocodingResult e Route.
esegece.sgcMaps.Vector	O leitor de Mapbox Vector Tile e o leitor de estilo MapLibre: VectorTile, VectorStyle, StyleEvaluator.
esegece.sgcMaps.Rendering	MapView, VectorTileSource, VectorTileRenderer e as classes de argumentos de eventos.
esegece.sgcMaps.WinForms	SgcMapControl.
esegece.sgcMaps.Offline	OsmReader, RouteGraphBuilder, OfflineRouter, GeoIndexBuilder e OfflineGeocoder.

As classes offline compartilham as classes de resultado dos serviços online: o OfflineRouter preenche uma **Route** e o OfflineGeocoder retorna objetos **GeocodingResult**, de modo que o código que exibe um resultado funciona com ambos. É por isso que os exemplos offline também usam esegece.sgcMaps.Services.

Classes a descartar

Estas classes guardam arquivos, conexões, threads ou objetos nativos do SkiaSharp, e implementam IDisposable. Descarte-as com Dispose, ou crie-as com uma instrução using:

- **MapView**: seus carregadores de tiles e as imagens dos ícones dos marcadores. O **SgcMapControl** descarta seu próprio MapView, seu buffer de fundo e seu timer.
- **Geocoder**, **Directions**, **Places**, **Elevation** e **MapTimeZone**: um HttpClient próprio, quando foram criados com um HttpMessageHandler.
- **TileDownloader**: seu HttpClient.
- **OfflineRouter** e **OfflineGeocoder**: Dispose chama Close, que libera o grafo de rota ou a conexão SQLite.

Veja Também

- Início Rápido: WinForms
- Renderização Sem Interface Visual

- Visão Geral (Delphi)
- Instalação do .NET
- Mapeamento de Classes
- Diferenças do .NET em Relação ao Delphi

Instalação do .NET

O instalador do sgcUI instala os pacotes Delphi, C++Builder e Lazarus. O sgcMaps for .NET, em vez disso, é distribuído como seus próprios pacotes NuGet, somente para a edição Registered.

Adicionando os pacotes

Adicione o pacote do host que você usa. Cada pacote traz aqueles de que depende, de modo que uma aplicação Windows Forms só precisa de esegece.sgcMaps.WinForms, e um renderizador sem interface visual só precisa de 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
```

Pacote	Quando adicionar	Target Frameworks
esegece.sgcMaps.WinForms	Um controle de mapa em um formulário Windows Forms.	net8.0-windows, net9.0-windows
esegece.sgcMaps.Rendering	Mapas renderizados sem janela, ou um mapa em um host próprio.	net8.0, net9.0
esegece.sgcMaps.Offline	Roteamento offline, geocodificação offline e leitura de arquivos OpenStreetMap.	net8.0, net9.0
esegece.sgcMaps	Somente o modelo de mapa, os formatos, o baixador de tiles e os serviços online, sem desenhar nada.	net8.0, net9.0

As mesmas linhas em um arquivo de projeto:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcMaps.WinForms" Version="2.0.0" />
  <PackageReference Include="esegece.sgcMaps.Offline" Version="2.0.0" />
</ItemGroup>
```

Quando os pacotes vêm como arquivos .nupkg em vez de um feed, adicione primeiro a pasta deles como uma fonte de pacotes:

```
dotnet nuget add source C:\Packages\sgcMaps --name sgcMaps
```

Requisitos

- .NET 8 ou .NET 9. Os pacotes não têm como alvo o .NET Framework nem o .NET Standard.
- O controle Windows Forms precisa do Windows. Os outros três pacotes rodam no Windows, no Linux e no macOS.
- O esegece.sgcMaps usa o **Microsoft.Data.Sqlite** para ler arquivos MBTiles e o índice GeoNames. Nada precisa ser instalado para isso.
- O esegece.sgcMaps.Rendering usa o **SkiaSharp** e traz o **SkiaSharp.NativeAssets.Linux.NoDependencies**, de modo que também renderiza em um servidor Linux ou em um contêiner, sem nenhuma biblioteca nativa extra.

Sem edição Community

Não há edição Community (gratuita) do sgcMaps for .NET, nem pacote com o sufixo **.Community**. Todos os pacotes acima exigem uma licença Registered.

Veja Também

- Início Rápido: WinForms
- Instalação (Delphi)
- Visão Geral do .NET
- Mapeamento de Classes
- Diferenças do .NET em Relação ao Delphi

Início Rápido: WinForms

SgcMapControl é o controle de mapa Windows Forms, o equivalente em .NET do TsgcVCLMap do Delphi. Ele acompanha o pacote **esegece.sgcMaps.WinForms**, para net8.0-windows e net9.0-windows, e traz consigo esegece.sgcMaps.Rendering e esegece.sgcMaps.
O controle hospeda um MapView, o motor de mapa, em sua propriedade **Map**. Ele pinta a visualização com SkiaSharp em um buffer de fundo e encaminha o mouse, a roda e o teclado para ela. Os tiles são carregados em threads em segundo plano, de modo que a interface do usuário nunca espera pela rede; o controle se repinta na thread da interface quando eles chegam, e dispara TileError também na thread da interface.

1. Criar o projeto

```
dotnet new winforms -n MapQuickStart
cd MapQuickStart
dotnet add package esegece.sgcMaps.WinForms --version 2.0.0
```

2. Exibir um mapa

Substitua Program.cs pelo programa abaixo e exclua Form1.cs e Form1.Designer.cs. Ele adiciona uma camada OpenStreetMap com cache em disco, centraliza o mapa em Madri e adiciona um marcador arrastável e uma linha.
MapProviders.ApplyProvider copia as configurações do provedor para a camada: TileSourceType, UrlTemplate, MaxZoom e Attribution. Uma camada cujo Provider não é Custom baixa os tiles do seu provedor mesmo sem a chamada, portanto a chamada só é necessária quando você quer ler ou alterar esses valores.

3. Executar

```
dotnet run
```

Arraste o mapa para deslocá-lo, gire a roda para dar zoom e arraste o marcador. O título do formulário mostra o último clique, a posição em que o marcador foi solto ou o tile que falhou.

Propriedades

Propriedade	Tipo	Descrição
Map	MapView	O motor de mapa. Métodos como AddMarker, AddPolyline, FitToMarkers, SetLayerVisible e RenderToPng são chamados nele.
CenterLatitude, CenterLongitude	double	O centro da visualização, em graus.
Zoom	int	O nível de zoom, limitado por Options.MinZoom e Options.MaxZoom.
Bearing	double	A rotação do mapa em graus no sentido horário a partir do norte. O usuário pode girar o mapa quando Options.RotationEnabled é true.
Options	MapOptions	O comportamento e as decorações da visualização: limites de zoom, TileSize, botões de zoom, barra de escala, atribuição, cor de fundo, arrastar, zoom, rotação e o seletor de camadas.

MarkerOptions	MarkerOptions	O tamanho, a cor e a fonte padrão dos marcadores, e as opções de clustering em MarkerOptions.Clustering.
Layers	MapItemCollection<Layer>	As camadas de tiles.
Markers, Polylines, Polygons, Circles, Rectangles, Labels, Heatmaps	MapItemCollection<T>	As sobreposições. Adicione um novo item com Add(), ou com os métodos Add de Map.

As propriedades usuais de Control, como Dock, Anchor, Enabled, Visible e TabStop, também estão disponíveis.

Eventos

Evento	Argumentos	Disparado quando
MapClick	MapClickEventArgs: Latitude, Longitude	O mapa é clicado.
MarkerClick	MapItemEventArgs<Marker>: Item	Um marcador é clicado.
MarkerDragStart, MarkerDrag, MarkerDragEnd	MapItemEventArgs<Marker>: Item	Um marcador com Draggable definido como true é arrastado.
ShapeEdited	MapItemEventArgs<MapItem>: Item	O usuário moveu, adicionou ou excluiu um vértice de uma polilinha ou polígono editável.
ClusterClick	ClusterClickEventArgs: Count, Latitude, Longitude, Handled	Um cluster de marcadores é clicado. Defina Handled como true para tratar o clique você mesmo.
LayerChanged	MapItemEventArgs<Layer>: Item	Uma camada é exibida ou ocultada, por exemplo pelo seletor de camadas.
ZoomChanged, BearingChanged	EventArgs	O nível de zoom ou a rotação muda.
TileError	TileErrorEventArgs: Layer, Zoom, X, Y, Error	Um tile não pode ser carregado, por exemplo com um erro HTTP do servidor de tiles.

Estes são eventos comuns do C#: vários manipuladores podem ser anexados com +=.

Veja Também

- Renderização Sem Interface Visual
- Demos do .NET
- Tiles e Camadas
- Visão Geral do .NET
- Instalação do .NET
- Mapeamento de Classes
- Diferenças do .NET em Relação ao Delphi

Renderização Sem Interface Visual

O **MapView**, no pacote **esegece.sgcMaps.Rendering**, é o motor de mapa do sgcMaps for .NET. O SgcMapControl o exibe em uma janela, mas ele não precisa de janela alguma: renderiza em um canvas do SkiaSharp, em uma imagem ou em um arquivo PNG. Uma aplicação de console, um serviço do Windows, um daemon Linux ou um servidor web pode, portanto, produzir mapas estáticos, com as mesmas camadas e sobreposições do controle. É o equivalente em .NET do uso sem interface visual dos controles Delphi descrito em Renderização de Mapas Estáticos.

Renderizando um arquivo PNG

O programa abaixo renderiza a Torre Eiffel no zoom 15 com um marcador e um círculo, e grava um arquivo PNG de 1024 x 768.
RenderToPng desenha a visualização, espera os tiles que ainda estão sendo carregados e a desenha de novo até que todos os tiles tenham chegado ou **tileTimeout** tenha passado (30 segundos quando não é informado). Depois da chamada, **LastRenderMissingTiles** informa quantos tiles faltaram na imagem. O tamanho da visualização é restaurado depois, de modo que um MapView pode renderizar imagens de vários tamanhos.

Renderizando para a memória

RenderToImage retorna um SKImage em vez de gravar um arquivo. Codifique-o no formato de que você precisa, por exemplo para responder a uma requisição web com bytes PNG:

Desenhando em seu próprio canvas

Render(SKCanvas canvas, int width, int height, bool includeControls) desenha a visualização uma vez em qualquer canvas do SkiaSharp e não espera. Os tiles que ainda não foram carregados entram em uma fila em segundo plano, e o evento **Invalidated** pede a próxima pintura quando eles chegam. Invalidated pode ser disparado em uma thread de trabalho, de modo que um host deve mudar para a sua própria thread de interface antes de pintar, como faz o SgcMapControl.

O que é desenhado

- As camadas de tiles, os rótulos dos tiles vetoriais, os mapas de calor, polígonos, retângulos, polilinhas, círculos, marcadores e rótulos, nessa ordem.
- Os botões de zoom, o seletor de camadas e a bússola somente quando **includeControls** é true. O valor padrão é false em RenderToImage e RenderToPng.
- A barra de escala e a atribuição quando **Options.ShowScaleBar** e **Options.ShowAttribution** são true, seja qual for o valor de includeControls. Mantenha a atribuição visível: os provedores de tiles a exigem.

Membros para uso sem interface visual

Membro	Descrição
SetSize(int width, int height)	Define o tamanho da visualização, usado por FitBounds, FitToMarkers e pelas conversões de coordenadas antes de uma renderização.
SetCenter(double latitude, double longitude), Zoom, Bearing	A visualização.
FitBounds, FitToMarkers(int paddingPx), FitToPolyline(Polyline polyline, int paddingPx)	Escolhem o centro e o zoom que mostram uma área, os marcadores ou uma polilinha.

AddMarker, AddPolyline, AddPolygon, AddCircle, AddRectangle, AddLabel	Adicionam sobreposições. AddMarker recebe o texto do popup e da dica de ferramenta.
Render, RenderToImage, RenderToPng	Desenham a visualização, como descrito acima.
WaitForTiles(TimeSpan timeout)	Bloqueia até que os tiles em fila sejam carregados ou o tempo limite expire, e retorna true quando nada está pendente.
PendingTileCount	O número de tiles sendo carregados em segundo plano.
LastRenderMissingTiles	O número de tiles que faltaram na última renderização.
DrawnMarkerCount, DrawnClusterCount, DrawnLabelCount	O que a última renderização desenhou.
AttributionText()	A atribuição das camadas visíveis, para imprimir ao lado da imagem quando ela não é desenhada nela.
LatLonToScreen, ScreenToLatLon	Convertem entre coordenadas e pixels da visualização.

Linux e macOS

O pacote Rendering tem como alvo net8.0 e net9.0, sem dependência do Windows, e traz a biblioteca nativa do SkiaSharp para Linux (SkiaSharp.NativeAssets.Linux.NoDependencies), que não precisa de fontconfig nem de outro pacote do sistema. Um renderizador sem interface visual pode, portanto, rodar em um contêiner Linux. Os tiles de um servidor precisam de acesso HTTPS de saída; os tiles de uma pasta ou de um arquivo MBTiles não precisam de nenhum.

Veja Também

- Demos do .NET
- Dados Offline do .NET
- 05 Static Render (Delphi)
- Visão Geral do .NET
- Instalação do .NET
- Mapeamento de Classes
- Diferenças do .NET em Relação ao Delphi

Dados Offline do .NET

O sgcMaps for .NET pode exibir um mapa, traçar rotas e geocodificar sem conexão de rede. Ele usa três tipos de dados locais: tiles em disco, um grafo de rota criado a partir do OpenStreetMap e um índice GeoNames. Todos são compatíveis em arquivo com a biblioteca Delphi, de modo que os dados criados com as ferramentas do Delphi podem ser usados aqui, e vice-versa.

Dados	Criado com	Usado por	Pacote
Pasta de tiles (z/x/y)	TileDownloader, ou qualquer ferramenta de tiles	Uma camada com TileSourceType.Folder	esegece.sgcMaps
Arquivo MBTiles, raster ou vetorial	Qualquer ferramenta de MBTiles	Uma camada com TileSourceType.MBTiles ou TileSourceType.VectorMBTiles	esegece.sgcMaps, esegece.sgcMaps.Rendering para tiles vetoriais
Grafo de rota (.sgcgraph)	RouteGraphBuilder	OfflineRouter	esegece.sgcMaps.Offline
Índice GeoNames (SQLite)	GeoIndexBuilder	OfflineGeocoder	esegece.sgcMaps.Offline

Tiles offline

Uma camada lê seus tiles de uma pasta ou de um arquivo MBTiles quando **TileSourceType** é Folder ou MBTiles e **TileSourcePath** aponta para ele. Os arquivos MBTiles vetoriais são desenhados com o estilo MapLibre em **VectorStyleFile**, ou com o estilo integrado quando ele está vazio. Veja Tiles Offline e Tiles Vetoriais.

Baixando um pacote de tiles

O **TileDownloader** baixa os tiles de uma área e de uma faixa de níveis de zoom para uma pasta z/x/y, e ignora os tiles que já estão lá. Ele não tem um UrlTemplate padrão de propósito: tile.openstreetmap.org e a maioria dos servidores públicos de tiles proíbem downloads em massa, portanto aponte-o para o seu próprio servidor ou para um provedor cujos termos os permitam, e mantenha DelayMS acima de zero.

Roteamento offline

O **RouteGraphBuilder** lê um extrato do OpenStreetMap, um arquivo .osm.pbf ou um arquivo XML .osm, e grava a rede viária como um arquivo .sgcgraph com um perfil de carro, de bicicleta e a pé. O **OfflineRouter** carrega esse arquivo e responde a rotas sem rede. A rota é uma **Route**, a mesma classe que o serviço Directions online retorna, com a distância em metros, a duração em segundos, os pontos, os trechos e os passos.

Membro	Descrição
RouteGraphBuilder.Build(string inputFile, string outputFile)	Constrói o grafo. Suas estatísticas estão em WayCount, NodeCount, MissingNodes, VertexCount, EdgeCount, ShapeCount e NameCount, e Progress informa os bytes lidos.

OfflineRouter.GraphFile, Open(), Close(), Active	O arquivo do grafo, e sua abertura ou fechamento. Active é somente leitura; o construtor que recebe um nome de arquivo apenas define GraphFile.
OfflineRouter.Profile	OfflineRouterProfile.Car (padrão), Bike ou Foot.
OfflineRouter.GetRoute(Route route, params MapCoord[] waypoints)	Traça a rota por dois ou mais pontos de passagem. Retorna false e define LastError quando não há rota. Uma sobrecarga recebe um IReadOnlyList<MapCoord>.

Construir o grafo de um país inteiro leva tempo e memória; construa-o uma vez e distribua o arquivo .sgcgraph. Veja Roteamento Offline para as fases da construção.

Geocodificação offline

O **GeoIndexBuilder** transforma um dump do GeoNames em um índice SQLite, e o **OfflineGeocoder** o pesquisa por nome ou por posição. Baixe um dump como cities1000.txt, mais admin1CodesASCII.txt e countryInfo.txt, do servidor de exportação do GeoNames. Os resultados são objetos **GeocodingResult**, como no Geocoder online.

Membro	Padrão	Descrição
GeoIndexBuilder.InputFile, DatabaseFile		O dump do GeoNames a ler e o índice a gravar.
GeoIndexBuilder.Admin1CodesFile, CountryInfoFile		Os arquivos que transformam códigos em nomes de regiões e países.
GeoIndexBuilder.FeatureClasses	PA	As feature classes do GeoNames a importar: P para lugares povoados, A para áreas administrativas.
GeoIndexBuilder.MinPopulation	0	Ignora os lugares com população menor.
GeoIndexBuilder.IncludeAlternateNames	true	Também indexa os nomes alternativos, como os nomes em outros idiomas.
GeoIndexBuilder.BatchSize	2000	Linhas gravadas por lote.
OfflineGeocoder.MaxResults	5	O número máximo de resultados.
OfflineGeocoder.CountryCodes		Códigos de país ISO alpha-2 separados por vírgula, como ES,PT, que restringem toda busca.
OfflineGeocoder.Geocode, ReverseGeocode, ReverseGeocodeCountry		Busca por nome, busca dos lugares mais próximos de uma posição e retorna o país de uma posição.

O OfflineGeocoder, como o OfflineRouter, é aberto com Open() e fechado com Close() ou Dispose().

Lendo arquivos do OpenStreetMap

O **OsmReader** lê em fluxo um arquivo .osm.pbf ou XML .osm e dispara **Node**, **Way** e **Relation** para cada elemento, com suas tags, e **Progress** com os bytes lidos. ReadNodes, ReadWays, ReadRelations, SkipNodeTags e WayFilter ignoram o que você não precisa. O RouteGraphBuilder é construído sobre ele, e ele serve para qualquer outra importação de dados do OpenStreetMap.

Veja Também

- Demos do .NET
- Geocodificação Offline
- Visão Geral do .NET
- Instalação do .NET
- Mapeamento de Classes
- Diferenças do .NET em Relação ao Delphi

Mapeamento de Classes

A tabela mapeia cada componente e classe Delphi do sgcMaps para o seu equivalente .NET. As classes .NET omitem o prefixo Tsgc e usam os nomes usuais do .NET; seus membros mantêm os nomes do Delphi sempre que o conceito é o mesmo.

Controles de mapa e modelo

Delphi	.NET	Pacote
TsgcVCLMap	SgcMapControl (Windows Forms), com o seu motor em SgcMapControl.Map	esegece.sgcMaps.WinForms
TsgcFMXMap	Sem equivalente. O MapView pode ser hospedado por qualquer framework que desenhe com SkiaSharp.	esegece.sgcMaps.Rendering
Um controle de mapa que renderiza sem janela, TsgcMapRasterizer	MapView: Render, RenderToImage, RenderToPng	esegece.sgcMaps.Rendering
TsgcMapLayer, TsgcMapMarker, TsgcMapPolyline, TsgcMapPolygon, TsgcMapRectangle, TsgcMapCircle, TsgcMapLabel, TsgcMapHeatmap	Layer, Marker, Polyline, Polygon (furos em PolygonHoles), Rectangle, Circle, Label, Heatmap	esegece.sgcMaps
TsgcMapLayers, TsgcMapMarkers e as demais coleções	MapItemCollection<T>	esegece.sgcMaps
TsgcMapOptions, TsgcMapMarkerOptions, TsgcMapClusterOptions	MapOptions, MarkerOptions, ClusterOptions	esegece.sgcMaps
TsgcMapCoord, TsgcMapHeatmapPoint	MapCoord, HeatmapPoint (readonly record structs)	esegece.sgcMaps

Tiles, provedores e formatos

Delphi	.NET	Pacote
TsgcMapTileProvider, TsgcMapTileSourceType	TileProvider, TileSourceType	esegece.sgcMaps
sgcMapApplyProvider, sgcMapGetProviderInfo, sgcMapProviderNames, sgcMapExpandTileURL, TsgcMapProviderInfo	MapProviders.ApplyProvider, GetInfo, ProviderNames, ExpandTileUrl, ProviderInfo	esegece.sgcMaps
As fontes de tiles de pasta, MBTiles e 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
As funções de sgcMaps_Geometry (sgcMapDistance, sgcMapInitialBearing, sgcMapDestinationPoint, sgcMapGeodesicInterpolate, sgcMapPointInPolygon, sgcMapSimplify, sgcMapNearestPointOnPolyline, sgcMapClusterPoints, sgcMapRenderHeatmap)	A classe estática MapGeometry: Distance, InitialBearing, DestinationPoint, GeodesicInterpolate, PointInPolygon, Simplify, NearestPointOnPolyline, ClusterPoints, RenderHeatmap	esegece.sgcMaps
A matemática de tiles de sgcMaps_Classes (sgcMapLatLonToTileF, sgcMapTileFToLatLon, sgcMapLatLonToPixel, sgcMapPixelToLatLon, sgcMapTileCount)	A classe estática TileMath: TryLatLonToTile, TryTileToLatLon, TryLatLonToPixel, TryPixelToLatLon, TileCount	esegece.sgcMaps

Serviços online e roteamento

Delphi	.NET	Pacote
TsgcMapService	MapServiceBase	esegece.sgcMaps
TsgcMapGeocoder	Geocoder, com GeocodingResult e GeocodingProvider	esegece.sgcMaps
TsgcMapDirections	Directions, com Route, RouteLeg, RouteStep, DirectionsProvider, TravelMode e PolylineCodec	esegece.sgcMaps
sgcMapAddRouteToPolylines	Route.AddToPolylines	esegece.sgcMaps
TsgcMapPlaces	Places, com Place e PlacesProvider	esegece.sgcMaps
TsgcMapElevation	Elevation, com ElevationResult e ElevationProvider	esegece.sgcMaps
TsgcMapTimeZone	MapTimeZone, com MapTimeZoneInfo e TimeZoneProvider	esegece.sgcMaps
TsgcMapRouteCalculator	RouteCalculator	esegece.sgcMaps

Offline

Delphi	.NET	Pacote
TsgcMapOfflineRouter	OfflineRouter, com OfflineRouterProfile	esegece.sgcMaps.Offline

TsgcMapRouteGraphBuilder, a ferramenta RouteGraphBuilder	RouteGraphBuilder	esegece.sgcMaps.Offline
TsgcMapOfflineGeocoder	OfflineGeocoder	esegece.sgcMaps.Offline
TsgcMapGeoIndexBuilder, a ferramenta GeoIndexBuilder	GeoIndexBuilder	esegece.sgcMaps.Offline
TsgcMapOSMReader, TsgcMapOSMTags, TsgcMapOSMMember	OsmReader, OsmTags, OsmMember	esegece.sgcMaps.Offline

Sem equivalente em .NET

- TsgcMapLocation, a posição do dispositivo.
- Os controles FireMonkey e Lazarus.
- As ferramentas de linha de comando GeoIndexBuilder e RouteGraphBuilder. Chame as classes de mesmo nome, ou use o OfflineDemo, que constrói os dois arquivos.

Veja Também

- Visão Geral do .NET
- Instalação do .NET
- Diferenças do .NET em Relação ao Delphi

Diferenças do .NET em Relação ao Delphi

A porta .NET mantém o modelo, o comportamento e os formatos de arquivo da biblioteca Delphi. O que muda é a forma da API, onde o .NET tem suas próprias convenções.

Nomes

- As classes omitem o prefixo Tsgc: TsgcMapGeocoder passa a ser Geocoder e TsgcVCLMap passa a ser SgcMapControl. O Mapeamento de Classes lista todas.
- As enumerações também omitem seus prefixos: mtpOpenStreetMap passa a ser TileProvider.OpenStreetMap, e mtsMBTiles passa a ser TileSourceType.MBTiles.
- Alguns membros seguem a capitalização do .NET, como UrlTemplate e NominatimUrl. A maioria mantém os nomes do Delphi.

Um motor, um controle

No Delphi, os controles de mapa guardam o mapa. No .NET, o mapa é um **MapView**, que funciona sem janela, e o **SgcMapControl** é um controle Windows Forms que hospeda um em sua propriedade Map. O controle expõe diretamente a visualização e as coleções (Zoom, Layers, Markers...), e métodos como AddMarker, FitToMarkers ou RenderToPng são chamados em SgcMapControl.Map.

Os serviços online são assíncronos

No Delphi, cada requisição de um serviço tem um método síncrono e um método Async que responde em um evento de resultado, como OnGeocodeResult. No .NET, cada requisição é um único método baseado em Task que retorna o resultado: GeocodeAsync e ReverseGeocodeAsync, GetRouteAsync, SuggestAsync e RetrieveAsync, GetElevationsAsync e GetElevationAlongPathAsync, GetTimeZoneAsync, e RouteCalculator.CalculateAsync. Todos recebem um CancellationToken opcional. Não há eventos de resultado, e CheckSynchronize não é necessário em uma aplicação de console.

- Uma requisição que falha lança uma **ServiceException**, com o status HTTP em StatusCode (0 quando não houve resposta HTTP). Ela também define LastError e dispara o evento Error, como OnError no Delphi.
- Uma requisição sem resultado não é um erro: a lista fica vazia, e o Geocoder define LastError como "No results".
- Os serviços compartilham um único HttpClient por padrão. Um construtor que recebe um HttpMessageHandler ou um HttpClient permite usar um proxy ou um manipulador de teste.
- O espaçamento, em todo o processo, das requisições ao Nominatim, ao servidor de demonstração do OSRM e ao OpenTopoData, de uma por segundo, é o mesmo do Delphi.

Eventos

Os eventos são eventos C# do tipo EventHandler<T>, como MapClick com MapClickEventArgs. Vários manipuladores podem ser anexados com += e removidos com -=, enquanto um evento Delphi guarda um único método. Os dados de um parâmetro de evento do Delphi são uma propriedade dos argumentos: e.Item para um marcador, e.Latitude e e.Longitude para um clique, e.Handled para interromper a ação padrão.

Cores

As cores são valores uint na ordem ARGB, 0xAARRGGBB, como MapColors.Blue (0xFF0000FF), onde o modelo Delphi usa TColor. **MapColors.FromTColor** e **MapColors.ToTColor** convertem entre os dois, e **MapColors.FromRgb** cria uma cor opaca.

Coordenadas e coleções

- MapCoord e HeatmapPoint são readonly record structs, criados com new MapCoord(latitude, longitude).
- As coleções de sobreposições são MapItemCollection<T>, uma Collection<T> com Add(), BeginUpdate, EndUpdate e um evento Changed.
- O ícone de um marcador é o conteúdo de um arquivo de imagem em Marker.Icon, um array de bytes, carregado com LoadIconFromFile ou LoadIconFromStream.

Roteador e geocodificador offline

OfflineRouter e OfflineGeocoder são abertos com **Open()** e fechados com **Close()** ou Dispose(). Sua propriedade Active é somente leitura, enquanto no Delphi definir Active como True os abre. OfflineRouter.GetRoute e OfflineGeocoder.Geocode são síncronos, como no Delphi.

Renderização

O motor desenha com SkiaSharp em todas as plataformas. RenderToPng grava arquivos PNG; para outro formato, codifique o SKImage de RenderToImage. Veja Renderização Sem Interface Visual.

Plataformas e versões

- Os pacotes têm como alvo o .NET 8 e o .NET 9, e o controle tem como alvo net8.0-windows e net9.0-windows. Não há build para o .NET Framework.
- A versão dos pacotes é 2.0.0, separada da versão do Delphi.
- Não há edição Community. Veja Instalação do .NET.
- TsgcMapLocation, o controle FireMonkey e as ferramentas de linha de comando não têm equivalente.

Veja Também

- Visão Geral do .NET
- Instalação do .NET
- Mapeamento de Classes

Demos do .NET

Três demos acompanham o sgcMaps for .NET, na pasta **.net\demos\Maps** do sgcUI. Cada demo é um pequeno projeto em estilo SDK para net8.0 ou net8.0-windows. Ela referencia os projetos do sgcMaps quando o código-fonte da biblioteca está ao lado (.net\source\Maps), e os pacotes NuGet da versão 2.0.0 caso contrário.

Demo	Tipo	Pacotes	Mostra
Console-RenderDemo	Console, net8.0	esegece.sgcMaps, esegece.sgcMaps.Rendering	Um mapa renderizado em um arquivo PNG sem janela, com marcadores e uma rota do serviço Directions online.
OfflineDemo	Console, net8.0	esegece.sgcMaps, esegece.sgcMaps.Offline	Roteamento e geocodificação offline, incluindo a construção do grafo de rota e do índice GeoNames.
WinForms-Demo	Windows Forms, net8.0-windows	esegece.sgcMaps.WinForms	O controle de mapa com duas camadas base, marcadores, formas editáveis, um círculo, um mapa de calor e os eventos.

ConsoleRenderDemo

A versão .NET da demo Delphi 05 Static Render. Sem argumentos, renderiza Madri no zoom 13 com a rota de carro da Puerta del Sol ao Parque do Retiro, obtida do servidor público OSRM, e grava **madrid.png** na pasta atual. Os tiles são armazenados em cache em uma pasta tile_cache ao lado do executável.

```
dotnet run --project ConsoleRenderDemo
dotnet run --project ConsoleRenderDemo -- --center 48.8584,2.2945 --zoom 15 --size 800x600 --provider carto --marker 48.858
```

Argumento	Descrição
--center lat,lon	O centro do mapa.
--zoom z	O nível de zoom, 13 por padrão.
--size WxH	O tamanho da imagem, 1024x768 por padrão.
--provider osm|opentopo|esri|carto	O provedor de tiles, osm por padrão.
--marker lat,lon	Adiciona um marcador. Pode ser repetido.
--route lat,lon;lat,lon[;...]	Pede ao servidor OSRM uma rota pelos pontos e a desenha.
--out file.png	O arquivo de saída, madrid.png por padrão.

A demo imprime o provedor, a visualização, a rota, o caminho completo do arquivo e o texto de atribuição. Ela precisa de acesso à internet para os tiles e a rota.

OfflineDemo

Traça rotas e geocodifica sem nenhuma chamada de rede. Ela constrói o grafo de rota a partir de um extrato do OpenStreetMap, ou reutiliza o arquivo .sgcgraph quando ele existe, e traça a rota entre dois pontos de carro. Quando os argumentos do GeoNames são informados, ela também constrói o índice GeoNames, ou o reutiliza, e geocodifica uma 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	Descrição
--pbf file.osm.pbf	O extrato do OpenStreetMap. Obrigatório.
--from lat,lon, --to lat,lon	O início e o fim da rota. Obrigatórios.
--graph file.sgcgraph	O grafo de rota. Por padrão, o nome do extrato com a extensão .sgcgraph.
--geonames, --admin1, --countryinfo	O dump do GeoNames, admin1CodesASCII.txt e countryInfo.txt.
--geocode "query"	O lugar a pesquisar.
--db file.sqlite	O índice GeoNames. Por padrão, o nome do dump com a extensão .sqlite.

Os extratos estão disponíveis nos servidores de download do OpenStreetMap, como o Geofabrik, e os arquivos do GeoNames no servidor de exportação do GeoNames. Veja Dados Offline do .NET.

WinFormsDemo

Um formulário com um **SgcMapControl** e uma barra de status. Ele mostra:

- Duas camadas base, OpenStreetMap e OpenTopoMap, com o seletor de camadas e a rotação habilitada.
- Marcadores arrastáveis em Madri e Barcelona, e um marcador em Valência.
- Uma polilinha editável de Madri a Barcelona passando por Saragoça, e um polígono editável.
- Um círculo de 40 km ao redor de Valência, e um mapa de calor de 200 pontos ponderados ao redor de Sevilha.
- Os eventos MarkerClick, MapClick, MarkerDragEnd, ShapeEdited, ZoomChanged e TileError, gravados na barra de status.

Quando o formulário é exibido, o mapa se ajusta aos marcadores. Os tiles são armazenados em cache na pasta sgcMaps_tiles da pasta temporária.

```
dotnet run --project WinFormsDemo
```

Veja Também

- Início Rápido: WinForms
- Renderização Sem Interface Visual
- Demos (Delphi)
- Visão Geral do .NET
- Instalação do .NET
- Mapeamento de Classes
- Diferenças do .NET em Relação ao Delphi

Solução de Problemas

Os problemas abaixo são os encontrados com mais frequência no sgcMaps, com sua causa e a correção.

O mapa não mostra tiles

- Trate **OnTileError**. Ele é disparado uma vez para cada tile que falha, com a camada, as coordenadas do tile e o erro, como HTTP 403 Forbidden. Um tile que o servidor não tem (HTTP 404) não é um erro: ele apenas é contado em **LastRenderMissingTiles**, como um tile ausente de uma pasta.
- Quando uma camada não pode ser aberta de forma alguma, o mapa desenha o motivo na visualização, depois do texto "Tile source not available". O motivo usual é "Tile source directory not found": a camada não tem Provider e seu TileSourceType ainda é mtsFolder, o padrão, com um TileSourcePath que não existe.
- Tiles de um servidor exigem o Delphi 2009 ou posterior, e o controle Lazarus exibe apenas tiles de pasta (veja Lazarus). MBTiles e MBTiles vetorial exigem o Delphi XE7; em versões mais antigas a camada informa que o tipo de fonte de tiles não está disponível neste compilador.
- Do Delphi 2009 ao XE7 os tiles são baixados com o Indy, e o https precisa das bibliotecas OpenSSL 1.0.2 (libeay32.dll e ssleay32.dll) ao lado do executável. Sem elas, o erro diz que as bibliotecas OpenSSL não puderam ser carregadas. A partir do Delphi XE8 o mapa usa THTTPClient e não precisa de DLL.
- Um tile que falhou não é armazenado, portanto o mapa o pede novamente na próxima pintura. OnTileError não é disparado duas vezes para o mesmo tile até que as camadas sejam abertas novamente.

```
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;
```

Chaves de API

- Os provedores de tiles MapTiler, Thunderforest e Stadia precisam de uma chave de API na propriedade **ApiKey** da camada. Sem chave, OnTileError informa "This tile provider requires an API key (ApiKey)"; com uma chave errada, o provedor responde HTTP 401 ou 403.
- Os serviços online não verificam a chave antes de enviar uma requisição. O erro do provedor chega em **LastError** (métodos síncronos) ou **OnError** (métodos assíncronos), como HTTP 403 com a explicação do provedor.
- TsgcMapTimeZone com GeoNames recebe seu nome de usuário do GeoNames em ApiKey, não uma chave.
- Nunca distribua uma chave em um demo ou em código-fonte público. Os demos do sgcMaps têm uma caixa de edição para a chave.

Políticas de uso e o user agent

Os servidores públicos gratuitos são mantidos por voluntários ou patrocinadores, e cada um tem uma política de uso. Uma aplicação que a descumpre pode ser bloqueada sem aviso.

- **tile.openstreetmap.org** permite apenas visualização interativa: sem download em massa, sem pré-carga de áreas, com um user agent identificador e uma atribuição visível. As requisições de tiles dos controles de mapa se identificam como sgcMaps/1.0. Para uma aplicação em produção use um provedor comercial, seu próprio servidor de tiles ou tiles offline.

- **TsgcMapTileDownloader** não tem URLTemplate padrão de propósito. Aponte-o para seu próprio servidor ou para um provedor cujos termos permitam downloads em massa, nunca para tile.openstreetmap.org, e mantenha DelayMS acima de zero (250 ms por padrão). Ele se identifica como sgcMapsDownloader/1.0.
- Os serviços online enviam sgcMaps/2.0 em **UserAgent** por padrão. Defina-o como o nome da sua aplicação: o Nominatim exige um user agent identificador.
- As requisições ao Nominatim e ao OpenTopoData são espaçadas pelo componente em uma por segundo, para o processo inteiro, como suas políticas pedem. Uma chamada síncrona feita a partir da thread principal pode, portanto, esperar até um segundo; use os métodos Async em uma interface de usuário.
- O servidor OSRM padrão de TsgcMapDirections é o servidor de demonstração público: apenas uso leve, uma requisição por segundo e somente o perfil de carro. Use seu próprio servidor OSRM ou outro provedor para caminhada e ciclismo, e chame **ValidateRequest** ou **IsTravelModeSupported** para verificar um modo de viagem em relação ao provedor.

Os eventos assíncronos nunca chegam

Os métodos Async dos serviços executam a requisição em uma thread e disparam o evento de resultado, ou OnError, na thread principal com TThread.Queue. Uma aplicação VCL ou FireMonkey entrega esses eventos a partir de seu laço de mensagens. Uma aplicação de console, um serviço do Windows ou uma thread não tem esse laço, portanto deve chamar **CheckSynchronize** até a resposta chegar, e o componente deve permanecer vivo até lá. Os objetos de resultado passados ao evento são liberados quando o evento retorna, portanto copie o que precisar dentro do manipulador.

```
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.
```

O **Timeout** dos serviços e da fonte de tiles é aplicado com THTTPClient a partir do Delphi 10.1 Berlin. No Delphi XE8 e no Delphi 10 Seattle ele não tem efeito.

FireDAC SQLite

- Arquivos MBTiles, arquivos MBTiles vetoriais e o índice de TsgcMapOfflineGeocoder são bancos de dados SQLite lidos com FireDAC, a partir do Delphi XE7. Sua edição do Delphi deve incluir o FireDAC com o driver SQLite, que o pacote de runtime do sgcMaps exige.
- As units criam por conta própria o vínculo com o driver FireDAC SQLite e abrem os arquivos somente para leitura, portanto o formulário não precisa de nenhum componente FireDAC. Ao distribuir, inclua o que o driver FireDAC SQLite precisa em cada plataforma de destino, como descreve a documentação do FireDAC da sua versão do Delphi.
- Um arquivo ausente gera "MBTiles file not found" ou "Geocoder index not found" com o caminho. Distribua os arquivos com a aplicação e monte os caminhos em tempo de execução, pois um caminho relativo depende da pasta atual.
- Um arquivo MBTiles com outro esquema gera tiles em branco e nenhum erro, portanto verifique um arquivo novo em um demo primeiro.
- Cada TsgcMapOfflineGeocoder tem uma conexão e deve ser usado por uma thread por vez. Dê a cada thread seu próprio componente. Ao contrário do geocodificador online, ele gera uma exceção quando o índice não existe, em vez de retornar False.

Pastas de cache

- Uma camada online guarda seus tiles em **CacheFolder**. Quando CacheFolder está vazio ela usa TileSourcePath, e quando ambos estão vazios não há cache em disco, apenas um cache de 256 tiles em memória.
- As pastas são criadas conforme necessário. Quando a pasta não pode ser gravada, os tiles não são armazenados em cache e nenhum erro é gerado: isso acontece em Program Files e na pasta da aplicação no Android e no iOS. Use uma pasta do usuário, como TPath.GetCachePath no mobile.
- Os arquivos de cache são nomeados apenas por zoom, x e y. Dê a cada camada, provedor e configuração Retina uma pasta própria, ou os tiles de uma camada serão desenhados em outra.
- Os tiles em cache não expiram. Exclua a pasta para baixá-los novamente, por exemplo após uma mudança de estilo no provedor.

Telas de alta DPI

O controle VCL desenha seus tiles com **Options.TileSize** pixels, 256 por padrão, e não se ajusta por conta própria ao DPI do monitor. Em uma tela de alta DPI, defina TileSize a partir dos pixels por polegada (de 64 a 1024) e defina **Retina** nas camadas, para que os provedores que o suportam enviem tiles de 512 pixels que permanecem nítidos quando ampliados. Uma camada vetorial com Retina desenha seus tiles com 512 pixels. Os provedores que suportam tiles retina estão listados em Tiles, Camadas e Provedores.

```
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;
```

Localização em dispositivos móveis

- No Android, habilite a permissão **Access fine location** nas opções do projeto e chame **RequestPermission** de TsgcMapLocation (Delphi 10.3 e posterior) antes de definir Active como True. Definir Active em tempo de design inicia o sensor antes de o usuário conceder a permissão.
- No iOS e no macOS, adicione NSLocationWhenInUseUsageDescription às informações de versão do projeto. No Windows, o serviço de localização deve estar habilitado nas configurações de privacidade.
- Os erros são "Location sensor not available", "Location sensor access denied or in error state" e "Location permission denied". Use **SimulateLocation** ou **LoadTrack** para testar sem um sensor, como faz o demo FireMonkey.

Roteamento offline

- **Open** de TsgcMapOfflineRouter gera "Invalid routing graph file", "Unsupported routing graph version" ou "Truncated routing graph file" quando o GraphFile não é um arquivo .sgcgraph completo desta versão. Construa-o novamente com a ferramenta RouteGraphBuilder.
- "Waypoint n is not near a routable road" significa que o ponto de passagem está fora da área do grafo, ou longe de qualquer via do perfil selecionado. "No route found between waypoints" significa que o grafo não tem conexão entre eles, o que acontece na borda de um extrato.
- O grafo inteiro é carregado em memória, portanto escolha o menor extrato que cobre sua área.
- O leitor do OpenStreetMap suporta arquivos .osm.pbf sem compressão e compactados com zlib, o formato dos serviços de extrato usuais. Um arquivo compactado com LZMA, LZ4, Zstandard ou bzip2 gera um erro.

Lazarus

O controle Lazarus exibe apenas tiles de uma pasta, porque não consegue baixar tiles, e não tem rotação nem opacidade. GeoJSON, GPX e KML não são compilados no FPC. Veja Lazarus.

Introdução

sgcDashboard é um dashboard nativo para Delphi e C++Builder, para VCL e FireMonkey. Um dashboard é uma grade de tiles. Cada tile é um controle filho que hospeda qualquer outro controle solto nele, de modo que um gráfico, um mapa, um visualizador de PDF, uma pré-visualização de relatório, um memo ou uma lista podem ficar lado a lado na mesma superfície temática. Um controle de cartão KPI mostra um único valor com sua tendência e um sparkline.
O sgcDashboard faz parte da suíte visual da eSeGeCe. Seus tiles hospedam os visualizadores do **sgcCharts** (gráficos de dados em tempo real), do **sgcMaps** (mapas), do **sgcPDF** (visualizador de PDF) e do **sgcReports** (pré-visualização de relatórios), e ele é desenhado com o **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro. O sgcDashboard exige o sgcUICore, que é instalado junto com ele. Os demais produtos são opcionais: um tile só os hospeda quando estão instalados.
Uma única propriedade **Theme** no dashboard alterna o dashboard, seus tiles e os controles hospedados entre claro e escuro. Uma versão Registered também permite que o usuário arraste e redimensione os tiles em tempo de execução e salva o layout como JSON, e a ponte **sgcDashboardWS** alimenta os cartões KPI a partir de uma conexão WebSocket, MQTT ou server-sent events do sgcWebSockets.

Edições

O sgcDashboard é distribuído como Registered, Trial e Community. A edição Community não inclui dois recursos: arrastar e redimensionar os tiles em tempo de execução com SaveLayout e LoadLayout, e a ponte de dados em tempo real sgcDashboardWS. Consulte Edições.

Por onde começar

- Arquitetura: o motor de layout, os tiles, o canvas e o tema do sgcUICore, VCL e FMX.
- Plataformas: as versões do Delphi, o C++Builder e as plataformas.
- Início Rápido VCL, Início Rápido FMX e Início Rápido C++Builder: um primeiro dashboard em poucos minutos.
- Componentes: os componentes de dashboard, tile, cartão KPI e fonte da ponte.
- Layout, Temas, Persistência, Dados em tempo real e Imagens KPI sem interface.
- Demos: os projetos de demonstração para Delphi e C++Builder.

Arquitetura

O sgcDashboard é construído em camadas. Cada camada depende apenas das que estão abaixo dela, de modo que o motor de layout e o renderizador de KPI compilam sem VCL nem FireMonkey.

Camada	Units	Função
Motor de layout	sgcDashboard_Layout	TsgcDashboardLayout: uma grade independente de framework que transforma a célula de cada tile (Col, Row, ColSpan, RowSpan) em um retângulo, com Columns, RowHeight, Gap e Padding em pixels independentes de dispositivo, posicionamento automático, resolução de colisões e o cálculo de encaixe para arrastar e redimensionar. Consulte Layout.
Dashboard e tiles	sgcDashboard_VCL, sgcDashboard_FMX	TsgcVCLDashboard e TsgcFMXDashboard possuem um motor de layout e posicionam seus tiles nele. TsgcVCLDashboardTile e TsgcFMXDashboardTile são controles filhos, como um TTabSheet dentro de um TPageControl, e um controle solto em um tile é hospedado por ele.
Cartão KPI	sgcDashboard_KPI, sgcDashboard_KPI_VCL, sgcDashboard_KPI_FMX	TsgcDashboardKPI é o modelo e o renderizador. TsgcVCLKPICard e TsgcFMXKPICard são controles enxutos que possuem um modelo e o desenham.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme, sgcUICore_JSON	O runtime compartilhado: o canvas vetorial e o rasterizador em que todo tile e todo cartão são desenhados, a base de controles VCL e FMX com ritmo de quadros (MaxFPS) e solicitações de repintura seguras para threads, os tokens dos temas claro e escuro e o aplicador de temas, e o leitor e gravador de JSON usados pela persistência do layout.
Ponte de dados em tempo real	sgcDashboardWS_Source, _WebSocket, _MQTT, _SSE	TsgcDashboardSource e seus descendentes leem um valor de cada mensagem JSON de um cliente sgcWebSockets e o enviam a um cartão KPI. A ponte é um pacote separado, sgcDashboardWS, que exige o sgcDashboard e o sgcWebSockets.

Os tiles são controles filhos

Um tile é um controle real dentro do dashboard. Em tempo de projeto, o verbo "Add Tile" do editor de componente do dashboard cria um tile, e qualquer controle é solto em um tile a partir da paleta. Mover ou redimensionar um tile no designer o encaixa na grade. Em tempo de execução, um tile é criado em código e seu **Parent** é definido como o dashboard. O tile desenha uma superfície de cartão com borda, cantos arredondados e um cabeçalho opcional com seu **Title**, e deixa sua área cliente para os controles que hospeda.

Pintura e temas

Os tiles e os cartões KPI são desenhados pelo canvas do sgcUICore, o mesmo canvas que o sgcCharts usa, de modo que um tema claro ou escuro tem a mesma aparência nos dois frameworks e na saída PNG e SVG sem interface. Um tile não desenha os controles que hospeda. Ele aplica o tema neles por meio de suas propriedades publicadas: consulte Temas.

VCL e FMX

As classes VCL e FMX têm as mesmas propriedades, métodos e eventos. O dashboard VCL descende de TScrollingWinControl e o dashboard FMX de TVertScrollBox, de modo que uma grade mais alta que a área visível rola na vertical. Todo tamanho do layout é em pixels independentes de dispositivo e é dimensionado pelo PPI do monitor na VCL e pela escala do formulário no FMX.

Veja Também

- Plataformas
- Layout
- Temas

Edições

O sgcDashboard é compilado a partir do mesmo código-fonte em todas as edições. A edição altera apenas quais recursos uma compilação inclui.

Edição	Inclui
Registered	Tudo: os dashboards e tiles VCL e FMX, os cartões KPI e as imagens KPI sem interface, os temas, o arrastar e redimensionar os tiles em tempo de execução com SaveLayout e LoadLayout, e a ponte de dados em tempo real sgcDashboardWS.
Trial	Os mesmos recursos da Registered. Uma compilação feita com SGC_TRIAL exibe um aviso de que é uma versão de avaliação depois que a janela de lançamento indicada na compilação tiver passado. Nada mais muda.
Community	Os dashboards e tiles, o layout em grade com posicionamento automático, os cartões KPI com sparklines e a saída PNG e SVG sem interface, e os temas claro e escuro. Não inclui o arrastar e redimensionar em tempo de execução com SaveLayout e LoadLayout, nem a ponte sgcDashboardWS.

O que a Community não inclui

A edição Community é compilada com SGC_DASHBOARD_PERSIST indefinido. Estes membros não existem em uma compilação Community:

- **AllowDrag**, **AllowResize** e **OnLayoutChanged** no dashboard.
- **SaveLayout**, **LoadLayout**, **SaveLayoutToFile**, **LoadLayoutFromFile**, **SaveLayoutToStream** e **LoadLayoutFromStream** no dashboard.
- **Locked** no tile VCL. No FMX, Locked é a propriedade padrão do FMX e continua disponível, apenas não há arrastar a bloquear.

A edição Community também é distribuída sem as units e os pacotes do sgcDashboardWS, portanto TsgcDashboardSource_WebSocket, TsgcDashboardSource_MQTT e TsgcDashboardSource_SSE não estão disponíveis. Um cartão KPI ainda pode ser alimentado por código com **Value** e **AddValue**.

A edição é uma escolha em tempo de compilação no sgcVer.inc (SGC_COMMUNITY, que também desativa SGC_TRIAL), não uma chave em tempo de execução. O código que usa um dos membros acima não compila em uma compilação Community, consulte Solução de Problemas.

Veja Também

- Persistência
- Dados em tempo real
- Plataformas

Instalação

O sgcDashboard faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcDashboard** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcDashboard

- **sgcUICore**, a base compartilhada dos produtos de layout e do sgcCharts. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcDashboard é selecionado.
- **sgcDashboard for sgcWebSockets**, listado em **sgcWebSockets bridges**. Instala os pacotes sgcDashboardWS com as fontes WebSocket, MQTT e SSE (veja Dados em Tempo Real). O instalador só o oferece quando o sgcWebSockets já está instalado na máquina e a sua licença inclui o sgcDashboard. Caso contrário, ele aparece esmaecido.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13`.
`Demos\Dashboard`	Os demos do sgcDashboard. Veja Demos.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcDashboard.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcDashboard, junto com sgcCharts, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard. As pontes sgcWebSockets não fazem parte do instalador Community, e o sgcPDF e o sgcMaps não fazem parte do Community. Veja Edições para saber o que a compilação Community deixa de fora.

Instalações autônomas anteriores

Se o instalador encontrar uma instalação autônoma mais antiga do sgcUICore, como a gravada pelo instalador autônomo do sgcCharts, ele lista os library paths e os pacotes de design dessa instalação na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcDashboardD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcDashboard<V>` / `dclsgcDashboard<V>`	Os componentes do sgcDashboard (runtime / design time).
`sgcDashboardWS<V>` / `dclsgcDashboardWS<V>`	A ponte sgcWebSockets (runtime / design time). Requer o sgcWebSockets.

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcDashboard`, depois `dclsgcDashboard`, e os pacotes da ponte por último. Em seguida, instale os pacotes de design time `dcl*` e adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcDashboardC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

.NET

Nada muda para o .NET. O sgcDashboard continua sendo distribuído como seus próprios pacotes NuGet (`esegece.sgcDashboard` e os pacotes complementares). Veja a instalação do .NET.

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	TsgcVCLDashboard, TsgcVCLDashboardTile e TsgcVCLKPICard no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	TsgcFMXDashboard, TsgcFMXDashboardTile e TsgcFMXKPICard a partir do Delphi XE6 (SGC_DASHBOARD_FMX), em Win32, Win64, macOS, iOS e Android.
Linux64	As units independentes de framework, o motor de layout (sgcDashboard_Layout) e o modelo KPI com sua saída PNG e SVG sem interface (sgcDashboard_KPI), a partir do Delphi 10.2 (SGC_DASHBOARD_LINUX). Os controles FMX no Linux exigem o FMX para Linux (FMXLinux).
Ponte de dados em tempo real	Os pacotes sgcDashboardWS, onde os pacotes do sgcWebSockets para a mesma IDE estão instalados.
.NET	Um port em C# com a mesma API para WinForms, WPF e Avalonia, a saída headless em PNG e SVG e a ponte do sgcWebSockets, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão Geral do .NET.

Pacotes

O sgcDashboard traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcDashboardD13.dpk para o Delphi 13 e sgcDashboardC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcDashboard	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, sgcUICore
sgcDashboardVCL	Runtime, VCL	sgcDashboard_VCL e sgcDashboard_KPI_VCL (TsgcVCLDashboard, TsgcVCLDashboardTile, TsgcVCLKPICard)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcDashboard
sgcDashboardFMX	Runtime, FireMonkey, a partir do XE6	sgcDashboard_FMX e sgcDashboard_KPI_FMX (TsgcFMXDashboard, TsgcFMXDashboardTile, TsgcFMXKPICard)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcDashboard
dclsgcDashboard	Tempo de projeto, VCL	Registra os componentes VCL na página **SGC Dashboard** da paleta e adiciona o verbo "Add Tile" ao editor de componente do dashboard.	designide, rtl, vcl, sgcDashboard, sgcDashboardVCL
dclsgcDashboardFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra os componentes FireMonkey na página **SGC Dashboard** da paleta, com o mesmo verbo "Add Tile".	designide, rtl, fmx, sgcDashboard,

			sgcDashboardFMX, dclsgcDashboard
sgcDashboardWS, dclsgcDashboardWS	Runtime e tempo de projeto, ponte do sgcWebSockets	A ponte de dados em tempo real, independente de framework.	rtl, sgcDashboard, sgcWebSockets

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcDashboardFMX exige o dclsgcDashboard, portanto instale primeiro o dclsgcDashboard.

Versões do IDE e plataformas de destino

Delphi	sgcDashboard	sgcDashboardVCL	sgcDashboardFMX	dclsgcDashboard, dclsgcDashboardFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcDashboard, sgcDashboardVCL	sgcDashboardFMX	dclsgcDashboard, dclsgcDashboardFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcDashboard)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcDashboard)
XE6 a 12	Win32, Win64	Win32, Win64	Win32

13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcDashboardC13.cbproj, sgcDashboardVCLC13.cbproj, sgcDashboardFMXC13.cbproj, dclsgcDashboardC13.cbproj e dclsgcDashboardFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXDashboard, TsgcFMXDashboardTile e TsgcFMXKPICard) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcDashboardVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcDashboard_FMX e sgcDashboard_KPI_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcDashboard é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa o motor de layout e o modelo KPI com sua saída PNG e SVG sem interface. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Capturar um formulário VCL com PaintTo

Para capturar um formulário VCL, ou um dashboard nele, em um bitmap com PaintTo, trave o canvas do bitmap em torno da chamada. PaintTo em um canvas destravado pode perder seu contexto de dispositivo quando um controle nativo do formulário trata uma mensagem durante a captura, e o bitmap sai em branco ou incompleto. Um formulário FireMonkey usa MakeScreenshot no 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;
```

Veja Também

- Edições
- Arquitetura
- Solução de Problemas

Início Rápido VCL

Em tempo de projeto

- Solte um **TsgcVCLDashboard** da página **SGC Dashboard** da paleta em um formulário e defina **Align** como alClient.
- Clique com o botão direito no dashboard e escolha **Add Tile**. Repita para cada tile. Um novo tile tem Col e Row definidos como -1, portanto o posicionamento automático o coloca no primeiro espaço livre da grade. Defina Col, Row, ColSpan e RowSpan no Object Inspector, ou mova e redimensione o tile no designer, para dar a ele uma célula fixa.
- Selecione um tile e solte um controle nele, por exemplo um TsgcVCLChart do sgcCharts ou um TMemo, e defina seu **Align** como alClient.
- Solte um **TsgcVCLKPICard** em outro tile, defina seu Align como alClient e preencha **Title**, **Value**, **Trend** e **TrendValue**. Defina **ShowHeader** desse tile como False, pois o cartão já exibe seu próprio título.
- Defina **Theme** do dashboard como utDark. O dashboard, os tiles, o cartão KPI e os controles hospedados passam para a paleta escura.

Em código

O mesmo dashboard criado em tempo de execução. O programa compila como está com `sgcUI\Source` no caminho de busca.

```
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;
```

Veja Também

- TsgcVCLDashboard
- TsgcVCLDashboardTile
- TsgcVCLKPICard
- Layout
- Temas

Início Rápido FMX

Os controles FireMonkey exigem o Delphi XE6 ou posterior. Eles são usados exatamente como os da VCL: solte um **TsgcFMXDashboard**, defina **Align** como Client, use o verbo **Add Tile** e solte controles ou um **TsgcFMXKPICard** nos tiles.

Em 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;
```

Veja Também

- TsgcFMXDashboard
- TsgcFMXDashboardTile
- TsgcFMXKPICard
- Plataformas

Início Rápido C++Builder

Instale os pacotes do sgcDashboard para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página **SGC Dashboard** da paleta exatamente como no Delphi. O verbo **Add Tile** funciona da mesma forma. O instalador adiciona os dois caminhos à IDE para você. Os cabeçalhos .hpp e os arquivos .lib gerados do sgcDashboard e do sgcUICore estão nessa única pasta, portanto o sgcUICore não precisa de um caminho próprio.

Em código

Veja Também

- Início Rápido VCL
- Demos
- Plataformas

Demos

As demos estão na pasta **Demos** da instalação, com um projeto Delphi em Demos\Delphi e a mesma demo para C++Builder em Demos\CBuilder. Cada pasta de demo tem um readme.html que explica o que a demo faz e como funciona. As demos são aplicativos VCL para Win32 e Win64.

Demo	Componentes	O que mostra
Basic	TsgcVCLDashboard, TsgcVCLDashboardTile, TsgcVCLKPICard	Um dashboard com seis tiles em uma grade de 12 colunas: quatro cartões KPI em tempo real movidos por um timer, um log de eventos e uma lista de servidores, com uma chave de tema claro e escuro que também aplica o tema ao TMemo e ao TListBox.
Suite	TsgcVCLDashboard, TsgcVCLKPICard, TsgcVCLChart, TsgcVCLMap, TsgcPDFViewerControl, TsgcReportPreviewVCL	Um dashboard cujos tiles hospedam os outros produtos VCL da eSeGeCe: um gráfico de linhas do sgcCharts em tempo real, um mapa do sgcMaps, um visualizador do sgcPDF e uma pré-visualização do sgcReports, ao lado de dois cartões KPI, todos seguindo o mesmo tema. Exige os códigos-fonte do sgcCharts, sgcMaps, sgcPDF e sgcReports no caminho de busca.
LiveKPI	TsgcVCLKPICard, TsgcDashboardSource_WebSocket, TsgcWebSocketClient, TsgcWebSocketServer	Três cartões KPI vinculados a um feed WebSocket por meio do TsgcDashboardSource_WebSocket. Um TsgcWebSocketServer dentro da demo envia por broadcast uma mensagem JSON a cada 500 ms, e cada fonte lê seu próprio ValuePath dela. Exige os códigos-fonte do sgcWebSockets e não faz parte da edição Community.
SaveLayout	TsgcVCLDashboard, TsgcVCLDashboardTile, TsgcVCLKPICard	Um dashboard que o usuário reorganiza em tempo de execução com AllowDrag e AllowResize, um layout salvo e carregado em layout.json com SaveLayoutToFile e LoadLayoutFromFile, e OnLayoutChanged. Não faz parte da edição Community.

Veja Também

- Início Rápido VCL
- Persistência
- Dados em tempo real

TsgcVCLDashboard

Unit: sgcDashboard_VCL
TsgcVCLDashboard é o dashboard VCL, somente para Windows. Ele organiza seus filhos TsgcVCLDashboardTile em uma grade de **Columns** colunas e linhas de **RowHeight** pixels, com **Gap** entre os tiles e **Padding** ao redor deles, e rola na vertical quando os tiles são mais altos que o controle. Os tiles são adicionados em tempo de projeto com o verbo **Add Tile** do editor de componente. Consulte Layout para saber como as células são posicionadas.

Propriedades

Propriedade	Tipo	Descrição
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. Desenha o fundo do dashboard e é seguido por todo tile cujo ParentTheme é True, que o aplica nos controles hospedados. Consulte Temas.
Columns	Integer	Número de colunas da grade. Padrão 12.
RowHeight	Integer	Altura de uma linha da grade, em pixels independentes de dispositivo. Padrão 120.
Gap	Integer	Espaço entre dois tiles, em pixels independentes de dispositivo. Padrão 12.
Padding	Integer	Espaço entre a borda do dashboard e os tiles, em pixels independentes de dispositivo. Padrão 12.
AutoFlow	Boolean	Quando True (padrão), um tile cujo Col ou Row é -1 preenche o primeiro espaço livre da grade. Quando False, -1 é lido como 0.
AllowDrag	Boolean	Move um tile em tempo de execução pelo seu cabeçalho. Padrão False. Não está na edição Community.
AllowResize	Boolean	Redimensiona um tile em tempo de execução pela alça em seu canto inferior direito. Padrão False. Não está na edição Community.
Layout	TsgcDashboardLayout (public, somente leitura)	O motor de layout da grade do dashboard. Consulte Layout.

Os membros usuais de TScrollingWinControl também são publicados: **Align**, **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop**, **Visible**, **OnClick**, **OnDblClick**, **OnMouseDown**, **OnMouseMove**, **OnMouseUp** e **OnResize**.

Métodos

Método	Retorno	Descrição
Arrange		Posiciona os tiles na grade agora. O dashboard se reorganiza quando um tile ou uma propriedade da grade muda, portanto raramente é necessário chamá-lo.

ApplyTheme		Aplica novamente o tema de cada tile aos controles que ele hospeda, por exemplo depois que controles foram adicionados ou alterados em código.
SaveLayout	string	As métricas da grade e a célula e a visibilidade de cada tile nomeado, como JSON. Não está na edição Community.
LoadLayout(const aJSON: string)		Aplica um texto de SaveLayout. Os tiles são associados por Name, nomes desconhecidos são ignorados e os tiles ausentes do texto mantêm suas células. Uma reorganização, depois OnLayoutChanged. Não está na edição Community.
SaveLayoutToFile(const aFileName: string)		Grava SaveLayout em um arquivo, UTF-8 sem BOM. Não está na edição Community.
LoadLayoutFromFile(const aFileName: string)		Lê um arquivo de layout gravado por SaveLayoutToFile. Um BOM é ignorado. Não está na edição Community.
SaveLayoutToStream(aStream: TStream)		Grava SaveLayout em um stream, UTF-8 sem BOM. Não está na edição Community.
LoadLayoutFromStream(aStream: TStream)		Lê um layout de um stream. Um BOM é ignorado. Não está na edição Community.

Eventos

Evento	Tipo	Descrição
OnLayoutChanged	TNotifyEvent	Um arraste ou redimensionamento em tempo de execução alterou a célula de um tile, ou LoadLayout foi aplicado. Não está na edição Community.

Exemplo

```
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;
```

Veja Também

- TsgcVCLDashboardTile
- TsgcFMXDashboard
- Layout
- Persistência
- Temas

TsgcVCLDashboardTile

Unit: sgcDashboard_VCL
TsgcVCLDashboardTile é um tile de um TsgcVCLDashboard. Ele desenha um cartão temático com borda, cantos arredondados e um cabeçalho opcional, e hospeda qualquer controle VCL solto nele: um gráfico, um mapa, um visualizador de PDF, uma pré-visualização de relatório, um cartão KPI ou um controle padrão. O tile aplica seu tema efetivo nos controles que hospeda. O tile nunca desenha os controles hospedados por conta própria.

Propriedades

Propriedade	Tipo	Descrição
Title	string	Texto do cabeçalho do tile.
ShowHeader	Boolean	Desenha a faixa do cabeçalho com o Title. Padrão True. A área cliente do tile começa abaixo do cabeçalho.
Col	Integer	Coluna da grade do tile, a partir de zero. Padrão -1, posicionado pelo posicionamento automático.
Row	Integer	Linha da grade do tile, a partir de zero. Padrão -1, posicionado pelo posicionamento automático.
ColSpan	Integer	Número de colunas que o tile cobre. Padrão 3, limitado de 1 até Columns.
RowSpan	Integer	Número de linhas que o tile cobre. Padrão 2, no mínimo 1.
ParentTheme	Boolean	O tile e seus controles hospedados seguem o Theme do dashboard. Padrão True.
Theme	TsgcUITheme	Tema do tile. Defini-lo limpa ParentTheme. Fora de um dashboard, um tile com ParentTheme usa utLight.
Locked	Boolean	O tile nunca é movido nem redimensionado em tempo de execução, mesmo quando o dashboard permite. Padrão False. Não está na edição Community.
Dashboard	TsgcVCLDashboard (public, somente leitura)	O dashboard ao qual o tile pertence, nil quando seu pai não é um dashboard.

Os membros usuais de controle também são publicados: **Constraints**, **Enabled**, **Hint**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **Visible**, **OnClick**, **OnContextPopup**, **OnDblClick**, **OnMouseDown**, **OnMouseMove**, **OnMouseUp** e **OnResize**. Um tile oculto (Visible False) deixa sua célula livre.

Métodos

Método	Retorno	Descrição
GetCell	TsgcDashboardCell	A célula solicitada do tile, um record com Col, Row, ColSpan e RowSpan.

SetCell(const aCell: TsgcDashboardCell)		Define Col, Row, ColSpan e RowSpan de uma vez, com uma única reorganização do dashboard.

Eventos

Evento	Tipo	Descrição
OnResize	TNotifyEvent	O tile foi redimensionado, por uma reorganização do dashboard ou por um redimensionamento em tempo de execução.

Exemplo

Os tiles seguem o tema do dashboard, a menos que definam o seu próprio:

```
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;
```

Veja Também

- TsgcVCLDashboard
- TsgcFMXDashboardTile
- Layout
- Temas

TsgcFMXDashboard

Unit: sgcDashboard_FMX
TsgcFMXDashboard é o dashboard FireMonkey, disponível a partir do Delphi XE6. Ele tem os mesmos membros de grade, tema e persistência que TsgcVCLDashboard. Ele descende de TVertScrollBox, portanto os tiles rolam na vertical, e seu **Padding** é um Integer em pixels independentes de dispositivo que substitui o Padding do tipo TBounds dos controles FMX. O verbo **Add Tile** do editor de componente está disponível na IDE para Windows.

Propriedades

Propriedade	Tipo	Descrição
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. Desenha o fundo do dashboard e é seguido por todo tile cujo ParentTheme é True, que o aplica nos controles hospedados. Consulte Temas.
Columns	Integer	Número de colunas da grade. Padrão 12.
RowHeight	Integer	Altura de uma linha da grade, em pixels independentes de dispositivo. Padrão 120.
Gap	Integer	Espaço entre dois tiles, em pixels independentes de dispositivo. Padrão 12.
Padding	Integer	Espaço entre a borda do dashboard e os tiles, em pixels independentes de dispositivo. Padrão 12.
AutoFlow	Boolean	Quando True (padrão), um tile cujo Col ou Row é -1 preenche o primeiro espaço livre da grade. Quando False, -1 é lido como 0.
AllowDrag	Boolean	Move um tile em tempo de execução pelo seu cabeçalho. Padrão False. Não está na edição Community.
AllowResize	Boolean	Redimensiona um tile em tempo de execução pela alça em seu canto inferior direito. Padrão False. Não está na edição Community.
Layout	TsgcDashboardLayout (public, somente leitura)	O motor de layout da grade do dashboard. Consulte Layout.

Métodos

Método	Retorno	Descrição
Arrange		Posiciona os tiles na grade agora. O dashboard se reorganiza quando um tile ou uma propriedade da grade muda, portanto raramente é necessário chamá-lo.
ApplyTheme		Aplica novamente o tema de cada tile aos controles que ele hospeda, por exemplo depois que controles foram adicionados ou alterados em código.

SaveLayout	string	As métricas da grade e a célula e a visibilidade de cada tile nomeado, como JSON. Não está na edição Community.
LoadLayout(const aJSON: string)		Aplica um texto de SaveLayout. Os tiles são associados por Name, nomes desconhecidos são ignorados e os tiles ausentes do texto mantêm suas células. Uma reorganização, depois OnLayoutChanged. Não está na edição Community.
SaveLayoutToFile(const aFileName: string)		Grava SaveLayout em um arquivo, UTF-8 sem BOM. Não está na edição Community.
LoadLayoutFromFile(const aFileName: string)		Lê um arquivo de layout gravado por SaveLayoutToFile. Um BOM é ignorado. Não está na edição Community.
SaveLayoutToStream(aStream: TStream)		Grava SaveLayout em um stream, UTF-8 sem BOM. Não está na edição Community.
LoadLayoutFromStream(aStream: TStream)		Lê um layout de um stream. Um BOM é ignorado. Não está na edição Community.

Eventos

Evento	Tipo	Descrição
OnLayoutChanged	TNotifyEvent	Um arraste ou redimensionamento em tempo de execução alterou a célula de um tile, ou LoadLayout foi aplicado. Não está na edição Community.

Exemplo

```
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;
```

Veja Também

- TsgcFMXDashboardTile
- TsgcVCLDashboard
- Início Rápido FMX

TsgcFMXDashboardTile

Unit: sgcDashboard_FMX
TsgcFMXDashboardTile é um tile de um TsgcFMXDashboard, com os mesmos membros que TsgcVCLDashboardTile. Um tile cuja propriedade padrão **Locked** do FMX é True nunca inicia um arraste ou redimensionamento em tempo de execução.

Propriedades

Propriedade	Tipo	Descrição
Title	string	Texto do cabeçalho do tile.
ShowHeader	Boolean	Desenha a faixa do cabeçalho com o Title. Padrão True. A área cliente do tile começa abaixo do cabeçalho.
Col	Integer	Coluna da grade do tile, a partir de zero. Padrão -1, posicionado pelo posicionamento automático.
Row	Integer	Linha da grade do tile, a partir de zero. Padrão -1, posicionado pelo posicionamento automático.
ColSpan	Integer	Número de colunas que o tile cobre. Padrão 3, limitado de 1 até Columns.
RowSpan	Integer	Número de linhas que o tile cobre. Padrão 2, no mínimo 1.
ParentTheme	Boolean	O tile e seus controles hospedados seguem o Theme do dashboard. Padrão True.
Theme	TsgcUITheme	Tema do tile. Defini-lo limpa ParentTheme. Fora de um dashboard, um tile com ParentTheme usa utLight.
Locked	Boolean	A propriedade padrão do FMX. Um tile bloqueado nunca é movido nem redimensionado em tempo de execução. Padrão False.
Dashboard	TsgcFMXDashboard (public, somente leitura)	O dashboard ao qual o tile pertence, nil quando seu pai não é um dashboard.

Os membros usuais de controle do FMX também são publicados, entre eles **Anchors**, **Enabled**, **Hint**, **Margins**, **Opacity**, **PopupMenu**, **Visible**, e os eventos de mouse, de pintura e de redimensionamento.

Métodos

Método	Retorno	Descrição
GetCell	TsgcDashboardCell	A célula solicitada do tile, um record com Col, Row, ColSpan e RowSpan.
SetCell(const aCell: TsgcDashboardCell)		Define Col, Row, ColSpan e RowSpan de uma vez, com uma única reorganização do dashboard.

Eventos

Evento	Tipo	Descrição
OnResize	TNotifyEvent	O tile foi redimensionado.
OnResized	TNotifyEvent	O redimensionamento do tile foi concluído.

Exemplo

```
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;
```

Veja Também

- TsgcFMXDashboard
- TsgcVCLDashboardTile
- Temas

TsgcDashboardKPI

Unit: sgcDashboard_KPI
TsgcDashboardKPI é o modelo e renderizador independente de framework de um cartão KPI: um título, um valor grande, um ícone opcional, uma seta de tendência com seu texto, um texto de rodapé e um sparkline dos últimos MaxPoints valores. **Render** desenha o cartão em qualquer canvas do sgcUICore, de modo que o mesmo código desenha TsgcVCLKPICard, TsgcFMXKPICard e as imagens PNG e SVG sem interface. Ele compila em todas as plataformas, inclusive Linux64.
AddValue e ClearValues podem ser chamados de qualquer thread, os valores são protegidos por um lock, e OnChange é então chamado a partir dessa thread. Todas as outras propriedades devem ser definidas na thread principal.

Propriedades

Propriedade	Tipo	Descrição
Title	string	Legenda no topo do cartão.
Value	string	O valor grande do cartão, já formatado.
Icon	string	Um glifo de texto curto desenhado em um círculo de destaque, por exemplo '$' ou '%'. Vazio não desenha ícone.
Trend	TsgcDashboardKPITrend	ktNone (padrão), ktUp, ktDown ou ktNeutral. Desenha a seta de tendência, verde para alta e vermelha para baixa.
TrendValue	string	Texto ao lado da seta de tendência, por exemplo '+4.2 %'.
FooterText	string	Texto pequeno na parte inferior do cartão.
AccentColor	TsgcUIColor	Cor $AARRGGBB do círculo do ícone e do sparkline. 0 (padrão) usa a cor de destaque do tema.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Dentro de um tile de dashboard, o tile o define.
MaxPoints	Integer	Número de valores que o sparkline mantém. Padrão 50. O valor mais antigo é descartado quando um novo chega.
ValueCount	Integer (public, somente leitura)	Número de valores do sparkline.
Values[Index: Integer]	Double (public, somente leitura)	Um valor do sparkline, do mais antigo para o mais recente.

Métodos

Método	Retorno	Descrição
AddValue(aValue: Double)		Adiciona um valor ao sparkline. Seguro para threads. Valores NaN e infinitos são ignorados.
ClearValues		Remove todos os valores do sparkline.

GetValueArray	TsgcDashboardKPIValues	Uma cópia dos valores do sparkline, do mais antigo para o mais recente.
Render(aCanvas: TsgcUICanvas; aWidth, aHeight: Integer; aScale: Double = 1)		Desenha o cartão em aCanvas, com aWidth por aHeight pixels de dispositivo. aScale é o fator HiDPI aplicado a todos os tamanhos.
SaveToPNG(const aFileName: string; aWidth, aHeight: Integer; aScale: Double = 1)		Renderiza o cartão em um arquivo PNG. aWidth e aHeight são pixels lógicos, a imagem tem Round(aWidth * aScale) por Round(aHeight * aScale) pixels. Uma sobrecarga grava em um TStream.
SaveToSVG(const aFileName: string; aWidth, aHeight: Integer; aScale: Double = 1)		Renderiza o cartão em um arquivo SVG. Uma sobrecarga grava em um TStream.
Assign(Source: TPersistent)		Copia as propriedades e os valores do sparkline de outro TsgcDashboardKPI.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Uma propriedade ou o sparkline mudou. Chamado na thread que fez a alteração.

Exemplo

```
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;
```

Veja Também

- Imagens KPI sem interface
- TsgcVCLKPICard
- TsgcFMXKPICard

TsgcVCLKPICard

Unit: sgcDashboard_KPI_VCL
TsgcVCLKPICard é o cartão KPI VCL, somente para Windows. O controle possui um TsgcDashboardKPI, publica suas propriedades e o desenha por meio da base de controles VCL do sgcUICore. Uma alteração apenas agenda um novo quadro, portanto AddValue pode ser chamado de qualquer thread e o cartão é repintado na thread principal, no máximo MaxFPS vezes por segundo.

Propriedades

Propriedade	Tipo	Descrição
Title	string	Legenda no topo do cartão.
Value	string	O valor grande do cartão, já formatado.
Icon	string	Um glifo de texto curto desenhado em um círculo de destaque, por exemplo '$' ou '%'. Vazio não desenha ícone.
Trend	TsgcDashboardKPITrend	ktNone (padrão), ktUp, ktDown ou ktNeutral. Desenha a seta de tendência, verde para alta e vermelha para baixa.
TrendValue	string	Texto ao lado da seta de tendência, por exemplo '+4.2 %'.
FooterText	string	Texto pequeno na parte inferior do cartão.
AccentColor	TsgcUIColor	Cor $AARRGGBB do círculo do ícone e do sparkline. 0 (padrão) usa a cor de destaque do tema.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Dentro de um tile de dashboard, o tile o define.
MaxPoints	Integer	Número de valores que o sparkline mantém. Padrão 50. O valor mais antigo é descartado quando um novo chega.
KPI	TsgcDashboardKPI (public, somente leitura)	O modelo do cartão, também utilizável para a saída PNG e SVG sem interface, com o tamanho de sua escolha.
MaxFPS	Integer (public)	Número máximo de quadros por segundo do controle, herdado da base de controles do sgcUICore.
FrameCount	Cardinal (public, somente leitura)	Número de quadros renderizados desde que o controle foi criado.

Os membros usuais de controle também são publicados: **Align**, **Anchors**, **Color**, **Constraints**, **Enabled**, **ParentColor**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **Visible** e os eventos de clique, de mouse e de redimensionamento.

Métodos

Método	Retorno	Descrição

AddValue(aValue: Double)		Adiciona um valor ao sparkline. Seguro para threads. Valores NaN e infinitos são ignorados.
ClearValues		Remove todos os valores do sparkline.
ContentChanged		Agenda um novo quadro. Seguro para threads. O próprio cartão o chama a cada alteração.

Eventos

Evento	Tipo	Descrição
OnClick	TNotifyEvent	O cartão foi clicado.
OnResize	TNotifyEvent	O cartão foi redimensionado.

Exemplo

Atualizando um cartão com uma nova leitura:

```
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;
```

Veja Também

- TsgcDashboardKPI
- TsgcFMXKPICard
- Dados em tempo real

TsgcFMXKPICard

Unit: sgcDashboard_KPI_FMX
TsgcFMXKPICard é o cartão KPI FireMonkey, disponível a partir do Delphi XE6. Ele tem os mesmos membros que TsgcVCLKPICard e desenha o mesmo modelo TsgcDashboardKPI.

Propriedades

Propriedade	Tipo	Descrição
Title	string	Legenda no topo do cartão.
Value	string	O valor grande do cartão, já formatado.
Icon	string	Um glifo de texto curto desenhado em um círculo de destaque, por exemplo '$' ou '%'. Vazio não desenha ícone.
Trend	TsgcDashboardKPITrend	ktNone (padrão), ktUp, ktDown ou ktNeutral. Desenha a seta de tendência, verde para alta e vermelha para baixa.
TrendValue	string	Texto ao lado da seta de tendência, por exemplo '+4.2 %'.
FooterText	string	Texto pequeno na parte inferior do cartão.
AccentColor	TsgcUIColor	Cor $AARRGGBB do círculo do ícone e do sparkline. 0 (padrão) usa a cor de destaque do tema.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Dentro de um tile de dashboard, o tile o define.
MaxPoints	Integer	Número de valores que o sparkline mantém. Padrão 50. O valor mais antigo é descartado quando um novo chega.
KPI	TsgcDashboardKPI (public, somente leitura)	O modelo do cartão, também utilizável para a saída PNG e SVG sem interface.
MaxFPS	Integer (public)	Número máximo de quadros por segundo do controle, herdado da base de controles do sgcUICore.

Os membros usuais de controle do FMX também são publicados, entre eles **Align**, **Anchors**, **Margins**, **Opacity**, **RotationAngle**, **Scale**, **Visible**, e os eventos de mouse, de pintura e de redimensionamento.

Métodos

Método	Retorno	Descrição
AddValue(aValue: Double)		Adiciona um valor ao sparkline. Seguro para threads. Valores NaN e infinitos são ignorados.
ClearValues		Remove todos os valores do sparkline.

Eventos

Evento	Tipo	Descrição
OnClick	TNotifyEvent	O cartão foi clicado.
OnResize	TNotifyEvent	O cartão foi redimensionado.

Exemplo

```
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;
```

Veja Também

- TsgcDashboardKPI
- TsgcVCLKPICard
- Início Rápido FMX

TsgcDashboardSource

Unit: sgcDashboardWS_Source
TsgcDashboardSource é a classe base das fontes da ponte do sgcWebSockets que alimentam um cartão KPI: TsgcDashboardSource_WebSocket, TsgcDashboardSource_MQTT and TsgcDashboardSource_SSE. Toda mensagem que um descendente recebe é lida como JSON: o número em **ValuePath** é adicionado ao sparkline do cartão e se torna seu Value, formatado com **ValueFormat**. O Trend do cartão é derivado do valor anterior da mesma fonte: maior é ktUp, menor é ktDown, igual é ktNeutral. A ponte não faz parte da edição Community. Ela exige os pacotes sgcDashboardWS e o sgcWebSockets.
O uso de threads é explicado em Dados em tempo real.

Propriedades

Propriedade	Tipo	Descrição
Active	Boolean	Quando False, a fonte ignora todas as mensagens e restaura o manipulador anterior do cliente. Padrão True.
Card	TComponent	O cartão KPI a alimentar: um TsgcVCLKPICard ou um TsgcFMXKPICard, qualquer componente que implemente IsgcDashboardKPIHost. Outros componentes são rejeitados.
ValuePath	string	Caminho JSON do número em cada mensagem, por exemplo 'price' ou 'data.items[0].value'. Um número ou uma string numérica é aceito, com '.' como separador decimal, seja qual for a localidade. Vazio lê a mensagem inteira como o número.
ValueFormat	string	String de formato do texto Value do cartão, passada a Format com o número. Padrão '%.2f'. A saída usa o separador decimal da localidade do aplicativo.
TitlePath	string	Quando definido, copia esse valor JSON para o Title do cartão.
TrendValuePath	string	Quando definido, copia esse valor JSON para o TrendValue do cartão.

Eventos

Evento	Tipo	Descrição
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Um valor foi lido e o cartão foi atualizado. Thread principal no Delphi 2010 e posteriores.

Exemplo

```
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;
```

Veja Também

- Dados em tempo real
- TsgcVCLKPICard

TsgcDashboardSource_WebSocket

Unit: sgcDashboardWS_Source_WebSocket
TsgcDashboardSource_WebSocket alimenta um cartão KPI a partir das mensagens de texto de um TsgcWebSocketClient. Enquanto a fonte está Active e Client está atribuído, somente em tempo de execução, a fonte instala seu próprio manipulador OnMessage no cliente e mantém o manipulador que estava lá antes: cada mensagem de texto é primeiro lida no cartão e depois repassada a esse manipulador anterior, de modo que o manipulador do aplicativo continua sendo disparado. Definir Active como False, limpar Client ou destruir a fonte restaura o manipulador anterior, a menos que o aplicativo o tenha substituído nesse meio-tempo. A ponte não faz parte da edição Community. Ela exige os pacotes sgcDashboardWS e o sgcWebSockets.

Propriedades

Propriedade	Tipo	Descrição
Client	TsgcWebSocketClient	O cliente WebSocket cujas mensagens de texto são lidas.
Active	Boolean	Quando False, a fonte ignora todas as mensagens e restaura o manipulador anterior do cliente. Padrão True.
Card	TComponent	O cartão KPI a alimentar: um TsgcVCLKPICard ou um TsgcFMXKPICard, qualquer componente que implemente IsgcDashboardKPIHost. Outros componentes são rejeitados.
ValuePath	string	Caminho JSON do número em cada mensagem, por exemplo 'price' ou 'data.items[0].value'. Um número ou uma string numérica é aceito, com '.' como separador decimal, seja qual for a localidade. Vazio lê a mensagem inteira como o número.
ValueFormat	string	String de formato do texto Value do cartão, passada a Format com o número. Padrão '%.2f'. A saída usa o separador decimal da localidade do aplicativo.
TitlePath	string	Quando definido, copia esse valor JSON para o Title do cartão.
TrendValuePath	string	Quando definido, copia esse valor JSON para o TrendValue do cartão.

Métodos

Método	Retorno	Descrição
ProcessMessage(const aText: string)	Boolean	Lê uma mensagem como se viesse de Client. False quando não é um JSON válido ou ValuePath não é um número. Útil para testes e para mensagens que chegam por outro caminho.

Eventos

Evento	Tipo	Descrição

OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Um valor foi lido e o cartão foi atualizado. Thread principal no Delphi 2010 e posteriores.

Exemplo

```
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;
```

Veja Também

- TsgcDashboardSource
- TsgcDashboardSource_MQTT
- Dados em tempo real

TsgcDashboardSource_MQTT

Unit: sgcDashboardWS_Source_MQTT
TsgcDashboardSource_MQTT alimenta um cartão KPI a partir das publicações de um TsgcWSPClient_MQTT. Enquanto a fonte está Active e Client está atribuído, somente em tempo de execução, a fonte instala seu próprio manipulador OnMQTTPublish no cliente e mantém o manipulador que estava lá antes: cada publicação é primeiro lida no cartão e depois repassada a esse manipulador anterior, de modo que o manipulador do aplicativo continua sendo disparado. Definir Active como False, limpar Client ou destruir a fonte restaura o manipulador anterior, a menos que o aplicativo o tenha substituído nesse meio-tempo. Um aplicativo que atribui Client.OnMQTTPublish depois que a fonte foi conectada deve atribuí-lo antes de definir Client, ou alternar Active, para manter ambos.
Topic seleciona as publicações que são lidas, com os curingas MQTT + (um nível) e # (este e todos os níveis inferiores). Um Topic vazio lê todas as publicações. A fonte não faz assinatura, o aplicativo assina por meio do cliente MQTT. Um payload que é apenas um número é lido com um ValuePath vazio. A ponte não faz parte da edição Community. Ela exige os pacotes sgcDashboardWS e o sgcWebSockets.

Propriedades

Propriedade	Tipo	Descrição
Client	TsgcWSPClient_MQTT	O cliente MQTT cujas publicações são lidas.
Topic	string	Filtro de tópico MQTT das publicações a ler. Vazio lê todas as publicações.
Active	Boolean	Quando False, a fonte ignora todas as mensagens e restaura o manipulador anterior do cliente. Padrão True.
Card	TComponent	O cartão KPI a alimentar: um TsgcVCLKPICard ou um TsgcFMXKPICard, qualquer componente que implemente IsgcDashboardKPIHost. Outros componentes são rejeitados.
ValuePath	string	Caminho JSON do número em cada mensagem, por exemplo 'price' ou 'data.items[0].value'. Um número ou uma string numérica é aceito, com '.' como separador decimal, seja qual for a localidade. Vazio lê a mensagem inteira como o número.
ValueFormat	string	String de formato do texto Value do cartão, passada a Format com o número. Padrão '%.2f'. A saída usa o separador decimal da localidade do aplicativo.
TitlePath	string	Quando definido, copia esse valor JSON para o Title do cartão.
TrendValuePath	string	Quando definido, copia esse valor JSON para o TrendValue do cartão.

Métodos

Método	Retorno	Descrição
ProcessPublish(const aTopic, aText: string)	Boolean	Lê uma publicação como se viesse de Client. False quando o tópico não corresponde a Topic ou o payload não pode ser lido.

Eventos

Evento	Tipo	Descrição
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Um valor foi lido e o cartão foi atualizado. Thread principal no Delphi 2010 e posteriores.

Exemplo

```
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;
```

Veja Também

- TsgcDashboardSource
- TsgcDashboardSource_WebSocket
- Dados em tempo real

TsgcDashboardSource_SSE

Unit: sgcDashboardWS_Source_SSE
TsgcDashboardSource_SSE alimenta um cartão KPI a partir dos server-sent events de um TsgcHTTP1Client. O aplicativo inicia o stream com **Client.GetSSE(URL)**, que bloqueia até o fim do stream, portanto costuma ser chamado a partir de uma thread de trabalho. Enquanto a fonte está Active e Client está atribuído, somente em tempo de execução, a fonte instala seu próprio manipulador OnSSEMessage no cliente e mantém o manipulador que estava lá antes: cada bloco é primeiro lido no cartão e depois repassado a esse manipulador anterior, de modo que o manipulador do aplicativo continua sendo disparado. Definir Active como False, limpar Client ou destruir a fonte restaura o manipulador anterior, a menos que o aplicativo o tenha substituído nesse meio-tempo. O manipulador do aplicativo ainda pode cancelar o stream.
OnSSEMessage entrega os bytes brutos do stream à medida que chegam, portanto um bloco pode conter vários eventos ou parte de um. A fonte armazena os blocos em buffer, divide-os em eventos a cada linha em branco (fins de linha LF, CRLF ou CR), une as linhas data: de um evento com LF, ignora comentários e os demais campos, e lê os dados de cada evento por meio de ValuePath. Um evento sem dados é ignorado. A ponte não faz parte da edição Community. Ela exige os pacotes sgcDashboardWS e o sgcWebSockets.

Propriedades

Propriedade	Tipo	Descrição
Client	TsgcHTTP1Client	O cliente HTTP cujos server-sent events são lidos.
Active	Boolean	Quando False, a fonte ignora todas as mensagens e restaura o manipulador anterior do cliente. Padrão True.
Card	TComponent	O cartão KPI a alimentar: um TsgcVCLKPICard ou um TsgcFMXKPICard, qualquer componente que implemente IsgcDashboardKPIHost. Outros componentes são rejeitados.
ValuePath	string	Caminho JSON do número em cada mensagem, por exemplo 'price' ou 'data.items[0].value'. Um número ou uma string numérica é aceito, com '.' como separador decimal, seja qual for a localidade. Vazio lê a mensagem inteira como o número.
ValueFormat	string	String de formato do texto Value do cartão, passada a Format com o número. Padrão '%.2f'. A saída usa o separador decimal da localidade do aplicativo.
TitlePath	string	Quando definido, copia esse valor JSON para o Title do cartão.
TrendValuePath	string	Quando definido, copia esse valor JSON para o TrendValue do cartão.

Métodos

Método	Retorno	Descrição
ProcessChunk(const aText: string)	Integer	Lê os eventos completos de um bloco do stream como se viesse de Client. Um evento incompleto espera o próximo bloco. Retorna o número de eventos cujos dados produziram um valor.

ResetStream		Descarta um evento incompleto em buffer, por exemplo antes de um novo stream começar.

Eventos

Evento	Tipo	Descrição
OnValue	TsgcDashboardSourceValueEvent	procedure(Sender: TObject; aValue: Double). Um valor foi lido e o cartão foi atualizado. Thread principal no Delphi 2010 e posteriores.

Exemplo

```
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;
```

Veja Também

- TsgcDashboardSource
- TsgcDashboardSource_WebSocket
- Dados em tempo real

Layout

O dashboard posiciona seus tiles em uma grade. A grade e as células são as mesmas na VCL e no FMX, e são calculadas pelo motor de layout independente de framework TsgcDashboardLayout (unit sgcDashboard_Layout), exposto pela propriedade **Layout** do dashboard.

Métricas da grade

Propriedade	Padrão	Significado
Columns	12	Número de colunas. A largura da coluna é a largura do dashboard, menos Padding nos dois lados e os espaços entre colunas, dividida por Columns.
RowHeight	120	Altura de uma linha em pixels independentes de dispositivo. Um dashboard tem tantas linhas quantas seus tiles precisarem e rola na vertical.
Gap	12	Espaço entre dois tiles, na horizontal e na vertical.
Padding	12	Espaço entre a borda do dashboard e a grade.
AutoFlow	True	Coloca no primeiro espaço livre os tiles cujo Col ou Row é -1.

Todos os tamanhos são em pixels independentes de dispositivo. O dashboard VCL os dimensiona pelo PPI do monitor dividido por 96, o dashboard FMX pela escala do formulário.

Células dos tiles

Cada tile solicita uma célula com **Col**, **Row**, **ColSpan** e **RowSpan**. Col e Row começam em zero. Um novo tile tem Col e Row definidos como -1 e uma extensão de 3 colunas por 2 linhas. Antes de posicionar um tile, o motor ajusta sua célula: ColSpan de 1 até Columns, RowSpan para no mínimo 1, e Col para que o tile não ultrapasse a última coluna.

Regra de posicionamento e colisão

- **Passagem 1, células explícitas.** Os tiles com Col e Row definidos são posicionados na ordem dos controles. Quando a célula se sobrepõe a um tile já posicionado, o tile é empurrado para baixo uma linha por vez até que sua célula esteja livre. Suas colunas nunca mudam.
- **Passagem 2, posicionamento automático.** Enquanto AutoFlow for True, os tiles cujo Col ou Row é -1 preenchem o primeiro espaço livre, percorrendo as linhas de cima para baixo e as colunas da esquerda para a direita. Uma coordenada informada restringe a busca: um tile com Col 6 e Row -1 vai para a primeira linha livre na coluna 6.
- Quando AutoFlow é False, um Col ou Row de -1 é lido como 0 e o tile é posicionado na passagem 1.
- Um tile oculto (Visible False) não ocupa nenhuma célula.

A célula solicitada permanece em Col, Row, ColSpan e RowSpan. A célula posicionada, depois do ajuste e da resolução de colisões, está em **Layout.PlacedCell[Index]** e seu retângulo em **Layout.Rect[Index]**.

Exemplo

```
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;
```

Veja Também

- TsgcVCLDashboard
- TsgcVCLDashboardTile
- Persistência

Temas

A propriedade **Theme** do dashboard, do tipo TsgcUITheme do sgcUICore_Theme, seleciona **utLight** (padrão), **utDark** ou **utCustom**. O dashboard desenha seu fundo com o tema, e todo tile cujo **ParentTheme** é True o acompanha. Um tile pode definir seu próprio **Theme**, o que limpa ParentTheme. O tema é aplicado quando muda e quando um controle é inserido em um tile.

O que o aplicador de temas define nos controles hospedados

Um tile aplica seu tema em cada controle que hospeda por meio das propriedades publicadas que o controle expõe, sem dependência em tempo de compilação do produto hospedado. Vale a primeira regra que corresponder:

1. Uma propriedade publicada **Theme** gravável do tipo TsgcUITheme é definida. Isso vale para TsgcVCLKPICard e TsgcFMXKPICard.
2. Caso contrário, um objeto publicado **Chart** com esse Theme recebe o tema: **Chart.Theme** de um TsgcVCLChart ou TsgcFMXChart do sgcCharts.
3. Caso contrário, uma propriedade publicada **BackgroundColor** gravável recebe a cor de superfície do tema, por exemplo o visualizador do sgcPDF.
4. Caso contrário, uma propriedade publicada **Color** gravável recebe a cor de superfície, por exemplo o mapa do sgcMaps, um TMemo ou um TListBox.

Quando a regra 3 ou 4 é alcançada, a cor do texto do tema também é definida para que o texto continue legível em uma superfície escura: **Font.Color** na VCL, e **TextSettings.FontColor** ou uma **FontColor** publicada no FMX. Um objeto publicado **Viewer** de um controle hospedado recebe o tema pelas mesmas regras, um nível abaixo.
Com **utCustom**, o aplicador não toca em nenhum controle hospedado, de modo que o aplicativo mantém controle total sobre as cores deles. Chame **ApplyTheme** no dashboard para aplicar o tema novamente depois que controles forem alterados em código.

Limites conhecidos

- A pré-visualização do sgcReports não publica nenhuma propriedade de cor, portanto mantém suas próprias cores nos dois temas.
- Um controle que não publica nenhuma das propriedades acima não é alterado.
- Somente os controles filhos diretos de um tile recebem o tema, além de seu objeto Viewer. Os controles aninhados dentro de um painel do tile mantêm suas cores.

Exemplo

```
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;
```

Veja Também

- TsgcVCLDashboardTile
- TsgcVCLKPICard
- Demos

Persistência

Uma compilação Registered ou Trial permite que o usuário reorganize os tiles em tempo de execução e salva a disposição como JSON. Esses membros são compilados com SGC_DASHBOARD_PERSIST e não existem na edição Community.

Arrastar e redimensionar em tempo de execução

- **AllowDrag**: pressionar o botão esquerdo do mouse sobre o cabeçalho de um tile o move. Um tile com ShowHeader False não pode ser movido.
- **AllowResize**: uma alça é desenhada no canto inferior direito de cada tile, e arrastá-la redimensiona o tile.
- Enquanto o mouse se move, a célula de destino é desenhada abaixo dos tiles e o tile se encaixa na grade. Escape cancela o arraste.
- **Locked**: um tile bloqueado nunca é movido nem redimensionado. No FMX, esta é a propriedade Locked padrão.
- **OnLayoutChanged** é disparado depois que um arraste ou redimensionamento altera uma célula, e depois 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 e LoadLayout

SaveLayout retorna as métricas da grade e, para cada tile com Name, sua célula e sua visibilidade. **LoadLayout** aplica esse texto: os tiles são associados por Name, nomes desconhecidos são ignorados, e os tiles ausentes do texto mantêm suas células. As variantes de arquivo e de stream usam UTF-8 sem BOM e ignoram um BOM ao carregar.

```
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 é a saída de SaveLayout para um dashboard com dois tiles nomeados, formatada aqui para leitura. SaveLayout a grava em uma única linha.

```
{
  "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}
  ]
}
```

Chave	Significado
version	Versão do formato, 1.
columns, rowHeight, gap, padding	As métricas da grade do dashboard.
tiles	Um objeto por tile nomeado: name, col, row, colSpan, rowSpan e visible.

Veja Também

- TsgcVCLDashboard
- Layout
- Demos
- Edições

Dados em Tempo Real

Um cartão KPI é alimentado por código com **Value**, **Trend** e **AddValue**, ou por uma conexão em tempo real por meio da ponte sgcDashboardWS. As fontes da ponte leem um número de cada mensagem JSON de um cliente sgcWebSockets e o enviam a um cartão. A ponte não faz parte da edição Community.

Fonte	Cliente	Lê
TsgcDashboardSource_WebSocket	TsgcWebSocketClient	Cada mensagem de texto.
TsgcDashboardSource_MQTT	TsgcWSPClient_MQTT	Cada publicação cujo tópico corresponde a Topic.
TsgcDashboardSource_SSE	TsgcHTTP1Client	Os dados de cada server-sent event.

Caminhos e formato

- **ValuePath** é o caminho JSON do número, por exemplo price ou data.items[0].value. Um número ou uma string numérica é aceito, com . como separador decimal, seja qual for a localidade. Um ValuePath vazio lê a mensagem inteira como o número.
- **ValueFormat** formata o número para o texto Value do cartão com Format. O padrão é %.2f. O texto usa o separador decimal da localidade do aplicativo.
- **TitlePath** e **TrendValuePath**, quando definidos, copiam esse valor JSON para Title e TrendValue do cartão.
- O Trend do cartão acompanha o valor anterior da mesma fonte: maior é ktUp, menor é ktDown, igual é kt-Neutral.
- Uma mensagem que não é um JSON válido, ou cujo ValuePath não é um número, é ignorada.

Threads

- O número é adicionado ao sparkline imediatamente, na thread em que a mensagem chega. TsgcDashboardKPI.AddValue é seguro para threads.
- No Delphi 2010 e posteriores, as atualizações de texto (Value, Title, TrendValue, Trend) são aplicadas na thread principal por meio de TThread.Queue, seguidas de OnValue. O cliente pode usar qualquer modo de NotifyEvents.
- Aplicativos de console e serviços não têm um loop de mensagens que execute a fila, portanto devem chamar CheckSynchronize.
- O Delphi 7 a 2009 não têm TThread.Queue. Nesses casos, a atualização e OnValue são executados na thread em que a mensagem chegou, e um manipulador de OnValue deve se sincronizar antes de tocar em um controle.

Exemplo

```
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;
```

Veja Também

- TsgcDashboardSource
- TsgcVCLKPICard
- Demos

Imagens KPI sem Interface

TsgcDashboardKPI renderiza um cartão KPI sem janela e sem VCL nem FMX, de modo que um aplicativo de console, um serviço ou um servidor web pode produzir o mesmo cartão como imagem, por exemplo para um relatório, um e-mail ou uma página web. A unit compila em todas as plataformas, inclusive Linux64.

- **SaveToPNG** grava um arquivo ou stream PNG. aWidth e aHeight são pixels lógicos e aScale os multiplica: 320 por 160 na escala 2 gera uma imagem de 640 por 320.
- **SaveToSVG** grava um arquivo ou stream SVG do mesmo cartão.
- O cartão de um controle está disponível por meio de sua propriedade **KPI**, de modo que **Card.KPI.SaveToPNG** salva o que o controle exibe em qualquer tamanho.

Exemplo

```
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;
```

Veja Também

- TsgcDashboardKPI
- Plataformas

Visão Geral do .NET

sgcDashboard .NET é um port C# gerenciado da biblioteca Delphi sgcDashboard, não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcDashboard** e mantêm os nomes do Delphi, então **TsgcDashboardLayout**, **TsgcDashboardCell**, **TsgcDashboardKPI**, **TsgcDashboardKPITrend** e **IsgcDashboardKPIHost** se leem da mesma forma nas duas linguagens, e o dashboard, o tile e o cartão KPI de cada host seguem o padrão de nomes do Delphi. Eles são desenhados com o port .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe, então um dashboard tem a mesma aparência no WinForms, no WPF, no Avalonia e na saída headless em PNG e SVG.
Um dashboard organiza tiles em uma grade de colunas e linhas. Cada tile tem uma célula (uma coluna, uma linha e a extensão de ambas), um cabeçalho opcional com um título, e hospeda qualquer controle: um cartão KPI, um gráfico do sgcCharts, um visualizador do sgcPDF, uma pré-visualização do sgcReports ou uma caixa de texto comum. O tema do dashboard chega aos tiles e aos controles que eles hospedam. Na edição Registered o usuário pode arrastar e redimensionar os tiles em tempo de execução, o layout é salvo e carregado como JSON, e a ponte do sgcWebSockets alimenta os cartões KPI a partir de uma conexão ao vivo.

Pacotes

O port é dividido em seis pacotes NuGet, de modo que um projeto só traz o host que utiliza.

PackageId	Contém	Frameworks de destino
esegece.sgcDashboard	O motor de layout **TsgcDashboardLayout** com sua persistência JSON, o modelo KPI **TsgcDashboardKPI**, a enumeração de tendência e a saída SVG sem interface. Sem dependência nativa.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDashboard.Rendering	A saída PNG sem interface de um cartão KPI (**SaveToPNG**), por meio do SkiaSharp.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDashboard.WinForms	**TsgcWinFormsDashboard**, **TsgcWinFormsDashboardTile** e **TsgcWinFormsKPICard**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDashboard.Wpf	**TsgcWpfDashboard**, **TsgcWpfDashboardTile** e **TsgcWpfKPICard**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDashboard.Avalonia	**TsgcAvaloniaDashboard**, **TsgcAvaloniaDashboardTile** e **TsgcAvaloniaKPICard**, no Avalonia 11.3.	net8.0, net9.0
esegece.sgcDashboard.sgcWebSockets	A ponte de dados em tempo real: **TsgcDashboardSource**, **TsgcDashboardSource_WebSocket**, **TsgcDashboardSource_MQTT** e **TsgcDashboardSource_SSE**. Depende do pa-	netstandard2.0, net8.0, net9.0

	cote esegece.sgcWebSockets. Somente na edição Registered.	

Cada pacote de host depende do esegece.sgcDashboard.Rendering e do pacote sgcUICore do mesmo host, então basta adicionar o pacote do host.

Edições

Edição	IDs dos pacotes	Inclui
Registered	esegece.sgcDashboard, .Rendering, .WinForms, .Wpf, .Avalonia e .sgcWebSockets	Tudo: a grade e os tiles, os cartões KPI, os temas, a saída sem interface, arrastar e redimensionar em tempo de execução, **SaveLayout** e **LoadLayout**, **OnLayoutChanged** e a ponte do sgcWebSockets.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcDashboard.WinForms.Community. Não existe pacote de ponte Community.	A grade e os tiles, os cartões KPI, os temas e a saída sem interface. Arrastar e redimensionar em tempo de execução, a persistência do layout (**SaveLayout**, **LoadLayout** e os **SaveToJSON** e **LoadFromJSON** do motor) e a ponte ficam de fora.

É a mesma divisão das edições do Delphi, veja Edições.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	TsgcVCLDashboard, TsgcVCLDashboardTile e TsgcVCLKPICard	Windows, no .NET Framework 4.7.2 e no .NET 8 e 9.
WPF	TsgcFMXDashboard, TsgcFMXDashboardTile e TsgcFMXKPICard	Windows, no .NET Framework 4.7.2 e no .NET 8 e 9.
Avalonia	TsgcFMXDashboard, TsgcFMXDashboardTile e TsgcFMXKPICard	Windows, Linux e macOS, no .NET 8 e 9.
Sem interface	TsgcDashboardKPI e o motor de layout	Qualquer plataforma do .NET Standard 2.0, incluindo servidores Linux.

Os tiles do WPF e do Avalonia seguem o tile do FireMonkey: hospedam um elemento e gerenciam o espaço ao redor dele por meio de **ContentPadding**. O tile do WinForms segue o tile da VCL e hospeda qualquer quantidade de controles.

Veja Também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF

- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Renderização sem Interface no .NET
- Feeds em Tempo Real no .NET
- Demos do .NET

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, os pacotes core e de renderização vêm junto como dependências.

Edição Registered

```
dotnet add package esegece.sgcDashboard.WinForms
dotnet add package esegece.sgcDashboard.Wpf
dotnet add package esegece.sgcDashboard.Avalonia
```

Adicione a ponte para alimentar cartões KPI a partir do sgcWebSockets. Ela traz junto o pacote esegece.sgcWebSockets:

```
dotnet add package esegece.sgcDashboard.sgcWebSockets
```

Edição Community

```
dotnet add package esegece.sgcDashboard.WinForms.Community
dotnet add package esegece.sgcDashboard.Wpf.Community
dotnet add package esegece.sgcDashboard.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcDashboard.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie em um projeto os pacotes Registered ou os Community, nunca os dois: eles contêm os mesmos tipos no mesmo namespace. O código que usa **AllowDrag**, **AllowResize**, **SaveLayout**, **LoadLayout**, **OnLayoutChanged** ou a propriedade **Locked** de um tile não compila com os pacotes Community; proteja-o com `#if !SGC_COMMUNITY` quando um projeto compila as duas edições.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:dash="clr-namespace:esegece.sgcDashboard;assembly=esegece.sgcDashboard.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicativo desktop Avalonia. Mapeie o namespace no XAML com `xmlns:dash="using:esegece.sgcDashboard"`.
- Sem interface: o esegece.sgcDashboard sozinho basta para o motor de layout e para o SVG. Adicione o esegece.sgcDashboard.Rendering para PNG, que traz o SkiaSharp e seus recursos nativos para Windows e Linux.
- Ponte: o esegece.sgcDashboard.sgcWebSockets tem como destino netstandard2.0, net8.0 e net9.0, portanto funciona com todos os hosts, inclusive o .NET Framework 4.7.2.

Em C# todo host precisa de `using esegece.sgcDashboard;`, e de `using esegece.sgcUICore;` para **TsgcUITheme** e os demais tipos dos temas. A ponte também precisa de `using esegece.sgcWebSockets;`.

Veja Também

- Visão Geral do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia

- Feeds em Tempo Real no .NET
- Demos do .NET

Início Rápido WinForms

Este exemplo cria um formulário WinForms com um dashboard de três tiles: dois cartões KPI e um log de eventos em uma caixa de texto comum. Um timer alimenta o cartão de receita, e uma caixa de seleção alterna o dashboard entre o tema claro e o escuro.

Passos

- Crie um projeto WinForms `net8.0-windows` e adicione o pacote **esegece.sgcDashboard.WinForms**, veja Instalação pelo NuGet.
- Crie um **TsgcWinFormsDashboard**, encaixe-o (dock) e defina **Columns** e **RowHeight**. A grade sempre se ajusta à largura do dashboard e rola na vertical.
- Para cada tile crie um **TsgcWinFormsDashboardTile**, dê a ele um **Name**, um **Title** e sua célula (**Col**, **Row**, **ColSpan** e **RowSpan**), adicione o conteúdo aos **Controls** do tile e o tile aos **Controls** do dashboard. Um controle com dock preenche a área abaixo do cabeçalho.
- Defina **Value**, **Trend**, **TrendValue** e **FooterText** de um **TsgcWinFormsKPICard** e chame **AddValue** para adicionar um ponto ao seu sparkline.
- Defina **Theme** do dashboard: os tiles, os cartões e a caixa de texto o acompanham.

Exemplo

AllowDrag e **AllowResize** permitem que o usuário mova um tile pelo cabeçalho e o redimensione pelo canto inferior direito, ajustado à grade. Fazem parte da edição Registered; remova as duas linhas para os pacotes Community.

Salvar e restaurar o layout

SaveLayout retorna as métricas da grade e a célula e a visibilidade de cada tile com nome como JSON, e **LoadLayout** aplica esse texto novamente. **SaveLayoutToFile** e **LoadLayoutFromFile** fazem o mesmo com um arquivo, **SaveLayoutToStream** e **LoadLayoutFromStream** com um stream UTF-8. **OnLayoutChanged** é disparado depois que um arrasto muda uma célula e depois de um carregamento. A persistência do layout faz parte da edição Registered.
O programa escreve:

```
{"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
```

O JSON é o texto que o **SaveLayout** do Delphi escreve, então um layout salvo por um aplicativo Delphi carrega em um aplicativo .NET e vice-versa. Os tiles são associados por **Name** sem diferenciar maiúsculas de minúsculas; um tile sem nome não é salvo.

Veja Também

- Instalação pelo NuGet
- Início Rápido WPF
- Início Rápido Avalonia
- Classes do .NET
- Persistência
- TsgcVCLDashboard
- Demos do .NET

Início Rápido WPF

O mesmo dashboard no WPF. Os tiles e o que eles hospedam são declarados em XAML: os tiles são o conteúdo do dashboard, e cada tile recebe um elemento como conteúdo.

Passos

- Crie um projeto WPF `net8.0-windows` e adicione o pacote **esegece.sgcDashboard.Wpf**, veja Instalação pelo NuGet.
- Mapeie o namespace na janela com `xmlns:dash="clr-namespace:esegece.sgcDashboard;assembly=esegece.sgcDashboard.Wpf"`.
- Declare um **TsgcWpfDashboard** e, dentro dele, um **TsgcWpfDashboardTile** por tile, com seu **Title**, **Col**, **Row**, **ColSpan** e **RowSpan**. O conteúdo de um tile é um **TsgcWpfKPICard** ou qualquer outro elemento.
- Dê um `x:Name` aos tiles: é o nome que o **SaveLayout** armazena.
- Alimente os cartões e alterne **Theme** no code behind.

Exemplo

MainWindow.xaml:
MainWindow.xaml.cs:
O App.xaml é o de um novo projeto WPF, com `StartupUri="MainWindow.xaml"`. **AllowDrag** e **AllowResize** fazem parte da edição Registered; remova os dois atributos para os pacotes Community.

Veja Também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcFMXDashboard
- Demos do .NET

Início Rápido Avalonia

O mesmo dashboard no Avalonia, no Windows, no Linux e no macOS. Como no WPF, os tiles e seu conteúdo são declarados em XAML.

Passos

- Crie um projeto desktop Avalonia `net8.0` com **Avalonia.Desktop** e **Avalonia.Themes.Fluent** 11.3, e adicione o pacote **esegece.sgcDashboard.Avalonia**, veja Instalação pelo NuGet.
- Mapeie o namespace na janela com `xmlns:dash="using:esegece.sgcDashboard"`.
- Declare um **TsgcAvaloniaDashboard** e, dentro dele, um **TsgcAvaloniaDashboardTile** por tile. O conteúdo de um tile é um **TsgcAvaloniaKPICard** ou qualquer outro controle.
- Alimente os cartões e alterne **Theme** no code behind.

Exemplo

MainWindow.axaml:
MainWindow.axaml.cs:
Program.cs:
O tile gerencia seu próprio **Padding** a partir de **ContentPadding** e **ShowHeader**, então um **Padding** definido no XAML é substituído. Em uma tela de toque, um toque no cabeçalho de um tile arrasta o tile e um toque em qualquer outro lugar move a grade, veja Diferenças do .NET em relação ao Delphi.

Veja Também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcFMXDashboard
- Demos do .NET

Classes do .NET

Todas as classes estão no namespace **esegece.sgcDashboard**. As classes de host são controles finos sobre o motor e o modelo independentes de framework, que são compartilhados pelos três hosts e pela saída sem interface.

Classes de host

Host	Classe	Classe 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 independentes de framework

Tipo	Descrição
TsgcDashboardLayout	O motor da grade: **Columns**, **RowHeight**, **Gap**, **Padding** e **AutoFlow**, células dos tiles para retângulos (**Arrange**, **Rect**, **ContentHeight**), fluxo automático, empurrão para baixo em colisões, a matemática de encaixe de arrastar e redimensionar (**SnapMove**, **SnapResize**) e o JSON do layout (**SaveToJSON**, **LoadFromJSON**, edição Registered).
TsgcDashboardCell	Uma struct com **Col**, **Row**, **ColSpan** e **RowSpan**. `sgcDashboard_Layout.sgcDashboardCell` cria uma.
TsgcDashboardLayoutTile	O nome, a célula e a visibilidade de um tile no JSON do layout. Edição Registered.
TsgcDashboardKPI	O modelo e o renderizador do KPI, veja TsgcDashboardKPI.
TsgcDashboardKPITrend	`ktNone` , `ktUp` , `ktDown` ou `ktNeutral` .
IsgcDashboardKPIHost	Implementada pelos três controles de cartão KPI; a ponte aceita qualquer cartão que a implemente.

Dashboard

Os mesmos membros em **TsgcWinFormsDashboard**, **TsgcWpfDashboard** e **TsgcAvaloniaDashboard**.

Membro	Tipo	Descrição
Columns	int	Número de colunas da grade, 12 por padrão.
RowHeight	int	Altura de uma linha da grade em pixels independentes de dispositivo, 120 por padrão.
Gap	int	Espaço entre os tiles, 12 por padrão.
Padding	int	Espaço ao redor da grade, 12 por padrão. Oculta `Control.Padding` no WinForms.
AutoFlow	bool	Coloca os tiles sem célula (Col ou Row -1) nas primeiras células livres. True por padrão.
Theme	TsgcUITheme	Tema do dashboard, de seus tiles e dos controles que eles hospedam.
Layout	TsgcDashboardLayout	O motor da grade, somente leitura. Oculta o evento `Control.Layout` no WinForms.
Tiles	coleção	WPF e Avalonia: os tiles, a propriedade de conteúdo do XAML.
VerticalOffset	double	Avalonia: a posição de rolagem da grade.
AllowDrag	bool	Move um tile pelo cabeçalho em tempo de execução. Edição Registered.
AllowResize	bool	Redimensiona um tile pelo canto inferior direito em tempo de execução. Edição Registered.
OnLayoutChanged	TsgcUINotifyEvent	Disparado depois que um arrasto muda uma célula e depois de um carregamento. Edição Registered.
SaveLayout, LoadLayout	método	O layout como uma string JSON. Edição Registered.
SaveLayoutToStream, LoadLayoutFromStream	método	O layout como um stream UTF-8. Edição Registered.
SaveLayoutToFile, LoadLayoutFromFile	método	O layout como um arquivo. Edição Registered.
Arrange	método	Organiza os tiles novamente.
ApplyTheme	método	Aplica o tema novamente aos tiles e ao seu conteúdo, por exemplo depois que o conteúdo alterou suas próprias cores.

Tile

Os mesmos membros em **TsgcWinFormsDashboardTile**, **TsgcWpfDashboardTile** e **TsgcAvaloniaDashboardTile**.

Membro	Tipo	Descrição
Dashboard	dashboard	O dashboard que contém o tile, ou null.
Title	string	Texto do cabeçalho.
ShowHeader	bool	Mostra a faixa do cabeçalho, true por padrão.
Col, Row	int	A célula solicitada, -1 por padrão para que o fluxo automático posicione o tile.
ColSpan, RowSpan	int	Tamanho do tile em colunas e linhas, 3 e 2 por padrão.
GetCell, SetCell	método	Os quatro valores como um único **TsgcDashboardCell**.
ParentTheme	bool	Adota o tema do dashboard, true por padrão.
Theme	Tsg-cUITheme	Tema do tile quando **ParentTheme** é false.
Locked	bool	O tile nunca inicia um arrasto nem um redimensionamento. Edição Registered.
Content	elemento	WPF e Avalonia: o elemento hospedado, a propriedade de conteúdo do XAML. No WinForms o tile hospeda os controles de sua coleção **Controls**.
ContentPadding	int	WPF e Avalonia: espaço ao redor do conteúdo, 8 por padrão; a altura do cabeçalho é somada no topo enquanto **ShowHeader** é true.

Cartão KPI

TsgcWinFormsKPICard, **TsgcWpfKPICard** e **TsgcAvaloniaKPICard** encapsulam um TsgcDashboardKPI, disponível na propriedade **KPI**, e repetem seus membros: **Title**, **Value**, **Icon**, **Trend**, **TrendValue**, **FooterText**, **AccentColor**, **Theme**, **MaxPoints** e **AddValue**.

Veja Também

- Visão Geral do .NET
- Diferenças do .NET em relação ao Delphi
- Renderização sem Interface no .NET
- Feeds em Tempo Real no .NET
- Layout
- Temas

Diferenças do .NET em relação ao Delphi

O port .NET mantém os nomes, os padrões e as regras de layout do Delphi. As diferenças abaixo vêm dos frameworks de host.

Os tiles são contêineres filhos

Host	Como um tile hospeda seu conteúdo
WinForms	Um tile é um contêiner `Control`, como o tile da VCL: adicione qualquer quantidade de controles aos seus **Controls**. Eles são posicionados abaixo do cabeçalho com um recuo de 8 pixels, então um controle com `Dock = DockStyle.Fill` preenche a área de conteúdo. O tile carrega o ParentControlDesigner, de modo que um controle pode ser solto nele no designer.
WPF	Um tile recebe um elemento por meio de **Content**, a propriedade de conteúdo do XAML (`[ContentProperty("Content")]`), como o tile do FireMonkey. O dashboard é o pai lógico de seus tiles, e **Tiles** é sua propriedade de conteúdo, então os tiles são declarados como filhos do elemento dashboard.
Avalonia	O mesmo modelo do WPF, por meio do atributo `[Content]` em **Content** do tile e em **Tiles** do dashboard. O dashboard é o pai lógico e visual de cada tile, e o tile é o do seu conteúdo.

No WPF e no Avalonia o tile calcula seu **Padding** a partir de **ContentPadding** e **ShowHeader**, então um **Padding** definido no XAML é substituído. Um tile oculto: no WPF um tile `Hidden` mantém sua célula e um tile `Collapsed` sai da grade; no Avalonia um tile com `IsVisible` false sai da grade.

Propagação do tema para os controles hospedados

Quando o **Theme** do dashboard muda, quando um tile entra em um dashboard, quando um conteúdo é adicionado a um tile e em **ApplyTheme**, os controles hospedados adotam o tema do tile por meio dos helpers do sgcUICore `sgcUIApplyThemeWinForms`, `sgcUIApplyThemeWpf` e `sgcUIApplyThemeAvalonia`, os equivalentes dos `sgcUIApplyThemeVCL` e `sgcUIApplyThemeFMX` do Delphi. Assim, um gráfico do sgcCharts, um visualizador do sgcPDF ou uma pré-visualização do sgcReports em um tile alterna entre claro e escuro junto com o dashboard. `utCustom` deixa os controles hospedados intactos.

O JSON do layout fica no motor

SaveToJSON e **LoadFromJSON** do **TsgcDashboardLayout** escrevem e leem o texto do layout; os métodos **SaveLayout** e **LoadLayout** dos três hosts apenas coletam seus tiles e os chamam. O texto é o mesmo que o dashboard do Delphi escreve, então os layouts circulam livremente entre Delphi, WinForms, WPF e Avalonia. A chave de um tile é `Control.Name` no WinForms, `FrameworkElement.Name` (`x:Name`) no WPF e `StyledElement.Name` no Avalonia. Veja Renderização sem Interface no .NET para um exemplo sem host.

Padding e Layout no WinForms

TsgcWinFormsDashboard declara dois membros com `new`: seu **Padding** é o espaço ao redor da grade, um `int`, e oculta `Control.Padding`, e sua propriedade **Layout** retorna o motor da grade e oculta o evento `Control.Layout`. Faça um cast do dashboard para `Control` para acessar os membros ocultos.

Theme no Avalonia

Todo controle Avalonia já tem uma propriedade **Theme**, o `ControlTheme` de `StyledElement`. O dashboard, o tile e o cartão KPI declaram seu próprio **Theme** do tipo **TsgcUITheme** com `new`. Por uma referência do tipo sgc é o tema sgc; por uma referência tipada como `Control` ou `StyledElement` é o tema de controle do Avalonia.

Arrastar e mover por toque no Avalonia

Um contato de toque que se move mais de 8 unidades na vertical move a grade. Com **AllowDrag** ou **AllowResize**, um toque no cabeçalho ou no canto de redimensionamento de um tile inicia o arrasto do tile imediatamente e nunca move a grade, como no FireMonkey; um toque em qualquer outro lugar ainda move a grade. Um arrasto com o mouse nunca move a grade. No WPF o toque e a caneta são promovidos a entrada de mouse, e o WinForms usa apenas o mouse.

O contorno do arrasto

Enquanto um tile é arrastado ou redimensionado, a célula de destino é marcada na cor de destaque com 25 por cento de preenchimento. O WinForms a desenha abaixo dos tiles, como o dashboard da VCL. O WPF a desenha em uma camada de adorner e o Avalonia em um visual de overlay, ambos acima dos tiles, enquanto o placeholder do FireMonkey fica abaixo deles. **Escape** cancela o arrasto em todos os hosts, e no WPF e no Avalonia a perda da captura também o cancela.

Rolagem

A grade sempre se ajusta à largura e rola na vertical, meia linha por clique da roda em todos os hosts. O WinForms rola por meio de `ScrollableControl`. O WPF coloca a grade em um `ScrollViewer`. O Avalonia mostra seu próprio `ScrollBar` e expõe a posição como **VerticalOffset**, que move os tiles para cima e os recorta ao dashboard.

Eventos

OnLayoutChanged é uma propriedade do tipo delegate nomeado do Delphi **TsgcUINotifyEvent**, como nos outros produtos sgc .NET: atribua-a com `=`, ela guarda um único handler. As origens da ponte assinam os eventos do sgcWebSockets .NET com `+=` e removem apenas o seu próprio handler, então os handlers do aplicativo continuam sendo disparados, veja Feeds em Tempo Real no .NET.

Sem verbos de designer e sem tile de mapa

O verbo de editor de componente do Delphi "Add Tile" não tem equivalente no WinForms: adicione os tiles em código, ou declare-os em XAML no WPF e no Avalonia. O encaixe em tempo de projeto de um tile movido também não faz parte do port. A demo Suite do Delphi tem um tile do sgcMaps; o sgcMaps não tem port .NET, então a demo Suite do .NET tem um gráfico, um visualizador de PDF e uma pré-visualização de relatório, mas nenhum mapa.

Edições

Os pacotes Community são compilados com `SGC_COMMUNITY`. Arrastar e redimensionar em tempo de execução, **Locked**, a persistência do layout e **OnLayoutChanged** são removidos da compilação, e o pacote da ponte não é produzido, como o bloco `SGC_DASHBOARD_PERSIST` do Delphi. Proteja o código que os usa com `#if !SGC_COMMUNITY`.

Vinculação de dados

No WPF e no Avalonia, as propriedades visíveis ao usuário são propriedades de dependência e propriedades do Avalonia, portanto podem ser vinculadas, estilizadas e animadas, veja Vinculação de dados no .NET.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Renderização sem Interface no .NET
- Feeds em Tempo Real no .NET
- Layout
- Temas
- Persistência

Vinculação de dados no .NET

Nos hosts WPF e Avalonia, as propriedades visíveis ao usuário do dashboard, de seus tiles e do cartão KPI são propriedades de dependência (WPF) e propriedades do Avalonia, portanto podem ser vinculadas a um view model, definidas por um Style e animadas. Este tópico se aplica somente a WPF e Avalonia. Os controles WinForms não mudam: suas propriedades são propriedades CLR simples, como antes.

Propriedades vinculáveis

Cada propriedade da tabela tem um campo estático com o sufixo Property, por exemplo `TsgcWpfDashboardTile.ColProperty`. No WPF os campos são DependencyProperty. No Avalonia são StyledProperty, exceto Col, Row, ColSpan e RowSpan, que são DirectProperty. As propriedades marcadas com **TwoWay** são vinculadas em TwoWay por padrão, porque um arrastar ou redimensionar em tempo de execução e LoadLayout as alteram, de modo que uma vinculação sem Mode mantém o view model atualizado. As demais propriedades são vinculadas em OneWay por padrão.

Classe WPF	Classe Avalonia	Propriedades vinculáveis
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 e AllowResize não estão na edição Community. Padding é o espaço ao redor da grade em unidades independentes de dispositivo, um int. No Avalonia, Theme é o tema do sgcUICore (TsgcUITheme) e oculta o tema de controle do Avalonia de StyledElement, que continua acessível por meio de `StyledElement.ThemeProperty`.

Comportamento

- **Um arrastar grava a célula de volta.** Um arrastar ou redimensionar em tempo de execução, e LoadLayout, alteram a célula de um tile, e o tile copia a célula ajustada para Col, Row, ColSpan e RowSpan. Com a vinculação TwoWay padrão, o view model recebe a célula onde o tile foi parar.
- **Uma célula vinculada é uma alteração de layout.** Uma nova célula que chega por uma vinculação de dados é tratada como um arrastar, portanto OnLayoutChanged do dashboard dispara (não na edição Community). Definidas pelo código (os setters CLR ou SetCell), as propriedades da célula se comportam como antes.
- **Um valor rejeitado é gravado de volta.** Um valor que o dashboard ou o tile não aceita (Columns ou RowHeight menor que 1, um Gap ou Padding negativo, um span menor que 1, um ContentPadding negativo) não é aplicado, e o valor em uso é gravado de volta na propriedade e em uma origem TwoWay, no WPF e no Avalonia. O mesmo vale para um valor que o cartão KPI corrige (MaxPoints menor que 1, um texto null). No WPF a origem recebe o valor em uso depois que o dispatcher executa o trabalho pendente. No Avalonia a gravação de volta cobre um valor que a origem envia quando a vinculação já está em execução. Um valor que a origem já tem quando a vinculação inicia é corrigido no controle, mas não é gravado de volta na origem.
- **LoadLayout atualiza as propriedades.** Depois de LoadLayout, as métricas da grade e as células dos tiles são copiadas de volta para as propriedades, de modo que as vinculações veem o layout restaurado.
- **Styles.** Um setter de Style pode definir qualquer propriedade vinculável. Um Theme do dashboard definido por um Style alcança todo tile cujo ParentTheme é true.

Propriedades que continuam sendo propriedades CLR

Propriedade	Motivo
Tiles (dashboard)	A coleção de tiles, preenchida pelos tiles declarados em XAML, e não um valor para vincular.
Layout (dashboard)	O objeto do mecanismo de layout, somente leitura. Suas métricas de grade são vinculadas por meio de Columns, RowHeight, Gap, Padding e AutoFlow.
VerticalOffset (dashboard Avalonia)	A posição de rolagem da última passagem de layout, e não uma configuração.
Theme (tile)	Enquanto ParentTheme for true, o tile lê o tema do dashboard, e definir Theme limpa ParentTheme. Vincule o Theme do dashboard, ou ParentTheme, em vez disso.
Theme (cartão KPI)	Definido pelo tile que hospeda o cartão.

Exemplo

O view model abaixo é compartilhado pelos dois exemplos. Revenue está vinculado ao Value de um cartão KPI, e RevenueCol e RevenueRow estão vinculados a Col e Row do seu tile sem Mode, portanto são vinculados em TwoWay: quando o usuário arrasta o tile, o view model recebe a nova célula. Defina AllowDrag como True no dashboard para que o usuário possa arrastar os tiles.

WPF

MainWindow.xaml cria o view model como DataContext da janela. O code-behind e o App.xaml são os do modelo WPF.

Avalonia

MainWindow.axaml define `x:DataType` como o view model, de modo que as vinculações são vinculações compiladas. O code-behind chama InitializeComponent, e a classe App é a de Início Rápido Avalonia.

Veja Também

- Diferenças do .NET em relação ao Delphi
- Classes do .NET
- Início Rápido WPF
- Início Rápido Avalonia

Renderização sem Interface no .NET

A engine de layout e o cartão KPI não precisam de nenhum framework de interface, então um servidor, um aplicativo de console ou um teste pode posicionar tiles e renderizar cartões em arquivos de imagem. É o equivalente .NET de Imagens KPI sem Interface.

Métodos

Método	Pacote	Descrição
TsgcDashboardKPI.SaveToSVG(stream or file, width, height, scale)	esegece.sgcDashboard	Escreve o cartão como texto SVG, byte a byte o SVG do cartão Delphi.
TsgcDashboardKPI.SaveToPNG(stream or file, width, height, scale)	esegece.sgcDashboard.Rendering	Um método de extensão que escreve um PNG com fundo transparente. A imagem tem largura vezes escala por altura vezes escala pixels.
TsgcDashboardKPI.Render(canvas, width, height, scale)	esegece.sgcDashboard	Desenha o cartão em qualquer canvas do sgcUICore.
TsgcDashboardLayout.Arrange(width, scale)	esegece.sgcDashboard	Posiciona os itens; **PlacedCell**, **Rect** e **ContentHeight** retornam o resultado.
TsgcDashboardLayout.SaveToJSON, LoadFromJSON	esegece.sgcDashboard	O texto do layout dos dashboards. Edição Registered.

Exemplo

O programa escreve revenue.svg e revenue.png, com 560 por 400 pixels, 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
```

O tile orders pede a coluna 2, linha 0, que se sobrepõe ao tile revenue, então o motor o empurra para baixo, para a linha 2. **Cell** retorna a célula que um tile pediu, **PlacedCell** a célula que ele obteve; o JSON armazena as células solicitadas, como no Delphi.
Os pacotes sem interface rodam no Windows, no Linux e no macOS. No Linux, o esegece.sgcDashboard.Rendering traz os recursos nativos do SkiaSharp para Linux.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Imagens KPI sem Interface
- TsgcDashboardKPI

- Layout

Feeds em Tempo Real no .NET

O pacote esegece.sgcDashboard.sgcWebSockets alimenta cartões KPI a partir de uma conexão ao vivo do sgcWebSockets .NET, como a ponte sgcDashboardWS do Delphi, veja Dados em Tempo Real. Uma origem lê um número de cada mensagem JSON e o envia para um cartão: o sparkline recebe o ponto, e **Value**, **Title**, **TrendValue** e **Trend** são atualizados. **Somente na edição Registered**: o pacote da ponte não é distribuído na edição Community.

Origens

Origem	Cliente	Lê
TsgcDashboardSource_WebSocket	TsgcWebSocketClient	Cada mensagem de texto, por meio de **OnMessage**.
TsgcDashboardSource_MQTT	TsgcWSProtocol_MQTT_Client	Cada publish cujo tópico corresponde a **Topic**, com os curingas do MQTT `+` e `#`, por meio de **OnMQTTPublish**. A origem não faz subscribe; o aplicativo faz o subscribe por meio do cliente.
TsgcDashboardSource_SSE	TsgcHTTP1Client em **Client**, ou TsgcSSEClient em **SSEClient**	Os dados de cada server sent event, a partir dos chunks de **GetSSE** e **OnSSEMessage**, ou a partir dos eventos de **OnEvent**. **GetSSE** bloqueia até o stream terminar, então chame-o a partir de uma thread de trabalho.

Todas as origens compartilham os membros de TsgcDashboardSource:

Membro	Descrição
Card	Um controle de cartão KPI (qualquer **IsgcDashboardKPIHost**) ou um **TsgcDashboardKPI** sem interface. Qualquer outro objeto gera **EsgcDashboardSource**.
ValuePath	Caminho JSON do número, por exemplo `price` ou `data.items[0].value`. Um número ou uma string numérica é lido, com ponto como separador decimal. Vazio lê a mensagem inteira como o número.
TitlePath, TrendValuePath	Quando definidos, copiam esse valor JSON para **Title** e **TrendValue** do cartão.
ValueFormat	Uma string de formato do Delphi (Format) com um argumento float, como `%.2f` ou `%.1f ms`, formatada com a cultura atual. Vazio escreve o número como o FloatToStr do Delphi.
Active	True por padrão. False ignora todas as mensagens.
OnValue	Chamado com cada número lido, depois que o cartão foi atualizado.

Threads

O ponto é adicionado ao sparkline imediatamente, na thread em que a mensagem chega. Os textos e a tendência (para cima, para baixo ou neutra em relação ao valor anterior da origem) são aplicados por meio do `SynchronizationContext` que era o atual quando a origem foi criada, a thread de interface de um aplicativo WinForms, WPF ou Avalonia, então todo modo de **NotifyEvents** do cliente é seguro. Uma origem criada sem contexto de sincronização, em um aplicativo de console ou em um serviço, aplica a atualização na thread da mensagem.

Exemplo

Três cartões KPI em um dashboard WinForms, alimentados por um fluxo de cotações WebSocket, um tópico de sensor MQTT e um feed de server sent events. Os nomes de host são placeholders para os seus próprios servidores.

Experimente na interface de loopback

Este programa de console inicia um **TsgcWebSocketServer** no mesmo processo na interface de loopback, conecta a ele um **TsgcWebSocketClient** e alimenta um cartão sem interface a partir de três mensagens de broadcast, então não precisa de rede.
Ele imprime o valor formatado na cultura atual, aqui com vírgula como separador decimal, a tendência em relação ao valor anterior e o número de pontos do sparkline:

```
EUR/USD 1,0839, trend ktDown, 3 points
```

Diferenças em relação ao Delphi

- **OnMessage**, **OnMQTTPublish**, **OnSSEMessage** e **OnEvent** são eventos .NET. Uma origem adiciona seu handler com `+=` e remove apenas o seu próprio handler, enquanto o Delphi salva, substitui e restaura o handler do aplicativo. Os handlers do aplicativo sempre são disparados, na ordem de assinatura.
- Não existe FreeNotification no .NET. Quando o cartão é um `IComponent`, como um controle WinForms, a origem limpa **Card** quando o cartão é descartado. Para qualquer outro cartão defina **Card** como null antes que o cartão desapareça.
- **SSEClient** da origem SSE, que recebe um **TsgcSSEClient**, é um acréscimo do .NET.

Veja Também

- Visão Geral do .NET
- Instalação pelo NuGet
- Diferenças do .NET em relação ao Delphi
- Dados em Tempo Real
- TsgcDashboardSource
- TsgcDashboardSource_WebSocket
- TsgcDashboardSource_MQTT
- TsgcDashboardSource_SSE
- Demos do .NET

Demos do .NET

As demos .NET estão na pasta `demos` do produto e no zip de demos, com a solution `sgcDashboard_Demos.sln`. Cada demo tem um README.md.

Demo	Host	Descrição
Basic	WinForms	Seis tiles em uma grade de 12 colunas: quatro **TsgcWinFormsKPICard** alimentados por um timer com um passeio aleatório, um log de eventos e uma lista de servidores em controles comuns, e um combo Theme que alterna o dashboard entre claro e escuro.
BasicWpf	WPF	O mesmo dashboard declarado em XAML em um **TsgcWpfDashboard**.
BasicAvalonia	Avalonia	O mesmo dashboard em um **TsgcAvaloniaDashboard**, no Windows, no Linux e no macOS.
Suite	WinForms	Os produtos da eSeGeCe em um só dashboard: dois cartões KPI, um gráfico do sgcCharts com uma linha ao vivo, um visualizador do sgcPDF e uma pré-visualização do sgcReports em tiles, e um seletor de claro e escuro. Não há tile de mapa, porque o sgcMaps não tem port .NET.
LiveKPI	WinForms	Três cartões KPI alimentados por **TsgcDashboardSource_WebSocket** a partir de um **TsgcWebSocketServer** no mesmo processo em uma porta de loopback livre. Somente na edição Registered.
SaveLayout	WinForms	Arraste e redimensione os tiles, depois salve o layout em layout.json, carregue-o de volta ou redefina-o. Somente na edição Registered.

Com os pacotes Community, LiveKPI e SaveLayout ainda compilam: eles removem da compilação o código Registered e explicam o limite na barra de status.

Compilar e executar

As demos precisam do SDK do .NET 8 ou posterior. Execute uma demo a partir da sua pasta:

```
dotnet run
```

Dentro da árvore do produto as demos são compiladas contra os projetos da biblioteca. A partir do zip de demos elas restauram os pacotes NuGet, a edição Community por padrão. Defina **SgcDashboardEdition** para usar os pacotes Registered:

```
dotnet build -p:SgcDashboardEdition=Registered
```

Autoverificação

Toda demo tem uma opção `--test` que abre a janela fora da tela e a controla: Basic verifica os tiles e suas células, alimenta os cartões e alterna o tema, Suite verifica o gráfico, o visualizador de PDF e a pré-visualização do relatório em seus tiles, LiveKPI verifica se todo cartão recebeu valores, e SaveLayout salva, move um tile, carrega e redefine. A demo escreve uma linha PASS ou FAIL por verificação no arquivo e termina com o número de verificações que falharam, então 0 significa que todas as verificações passaram.

```
dotnet run -- --test result.txt
```

Veja Também

- Visão Geral do .NET
- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Feeds em Tempo Real no .NET
- Demos do Delphi

Solução de Problemas

Undeclared identifier SaveLayout, AllowDrag ou OnLayoutChanged

A edição Community é compilada sem SGC_DASHBOARD_PERSIST, portanto AllowDrag, AllowResize, OnLayoutChanged, os métodos SaveLayout e LoadLayout e a propriedade Locked da VCL não existem, e o código que os usa falha com E2003 Undeclared identifier. Um formulário salvo por uma compilação Registered que tenha AllowDrag ou AllowResize em seu .dfm gera uma exceção "Error reading" com "Property AllowDrag does not exist". Remova esses membros ou use uma compilação Registered ou Trial. Consulte Edições.

Unit sgcDashboardWS_Source_WebSocket ou sgcWebSocket não encontrada

A ponte de dados em tempo real é um pacote separado, sgcDashboardWS, que exige o sgcDashboard e o sgcWebSockets. Instale primeiro o sgcWebSockets para a mesma IDE, depois os pacotes sgcDashboardWS, e adicione a pasta de código-fonte ou de biblioteca do sgcWebSockets ao caminho de busca. A ponte não é distribuída na edição Community.

Um cartão KPI alimentado por uma fonte da ponte nunca muda em um aplicativo de console ou serviço

No Delphi 2010 e posteriores, os textos do cartão são aplicados na thread principal por meio de TThread.Queue. Um aplicativo de console ou um serviço não tem um loop de mensagens que execute a fila, portanto deve chamar CheckSynchronize regularmente. Os valores do sparkline são adicionados imediatamente em qualquer caso.

O Value de um cartão mostra uma vírgula como separador decimal

ValueFormat é aplicado com Format, que usa o separador decimal da localidade do aplicativo. O próprio número JSON é sempre lido com ponto. Altere ValueFormat, ou defina o valor a partir de OnValue com a sua própria formatação.

O dashboard FMX não compila para Linux64

Os controles FMX no Linux exigem o FMX para Linux (FMXLinux) instalado na IDE. Sem ele, somente as units independentes de framework compilam no Linux64: o motor de layout e o modelo KPI com sua saída PNG e SVG sem interface. Consulte Plataformas.

Um controle hospedado mantém suas cores quando o tema muda

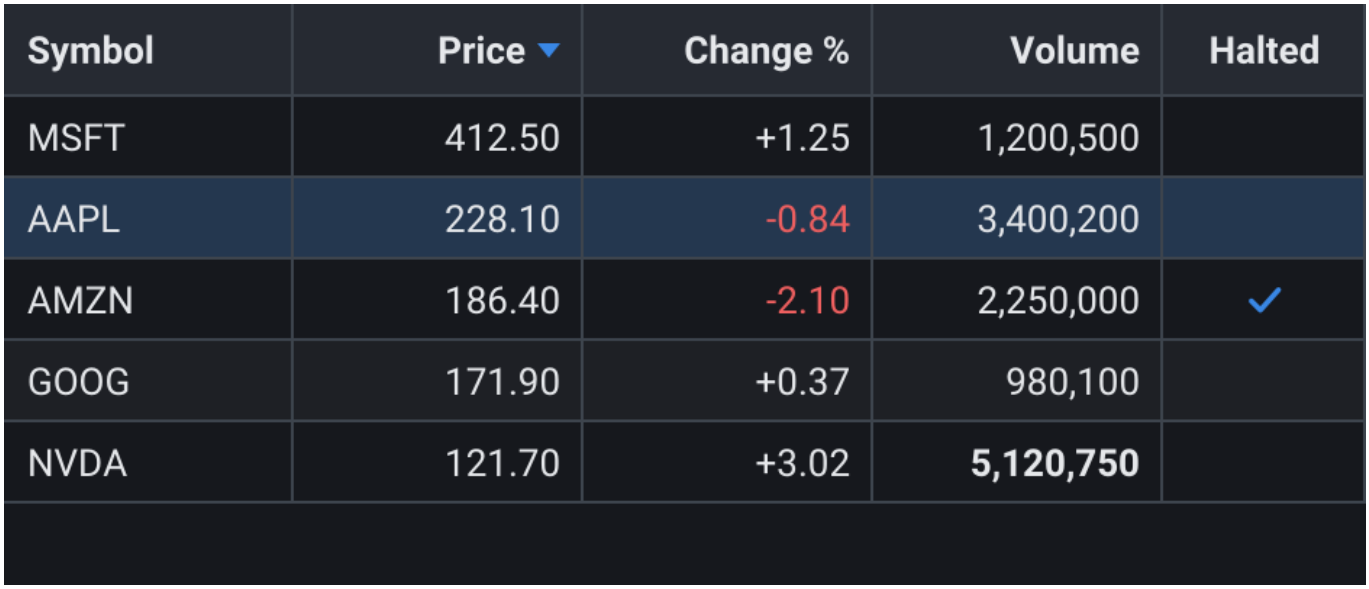
O tema só chega a um controle hospedado por meio de uma propriedade publicada Theme, Chart.Theme, BackgroundColor ou Color. A pré-visualização do sgcReports não publica nenhuma delas e mantém suas próprias cores, e os controles aninhados dentro de um painel de um tile não recebem o tema. Theme também é ignorado com utCustom. Consulte Temas.

A demo Suite ou LiveKPI não compila

A demo Suite exige os códigos-fonte do sgcCharts, sgcMaps, sgcPDF e sgcReports no caminho de busca, e a LiveKPI exige os códigos-fonte do sgcWebSockets. Consulte Demos.

Introdução

sgcGrid é uma grade de dados em tempo real para Delphi e C++Builder, para VCL e FireMonkey. Ela foi feita para dados que mudam o tempo todo: as linhas são atualizadas por chave a partir de qualquer thread, as células alteradas piscam em verde ou vermelho e esmaecem, rajadas de atualizações são agrupadas em repinturas temporizadas, e o modo de cauda mantém visíveis as linhas mais novas de um log ou de um feed. **TsgcVCLGrid** e **TsgcFMXGrid** ordenam por várias colunas, filtram, selecionam linhas ou células, redimensionam colunas, aplicam regras de formatação condicional e editam células no local.
O sgcGrid não é uma grade de banco de dados de uso geral, com bandas, agrupamentos e tabelas dinâmicas. É uma grade focada, vendida pelo desempenho: ela pinta apenas as linhas visíveis, e um quadro completo em 1080p de uma grade de 100.000 linhas leva poucos milissegundos. Consulte Desempenho.
O sgcGrid faz parte da suíte visual da eSeGeCe. Ele é desenhado com o **sgcUICore**, o runtime compartilhado com o canvas, o rasterizador e os temas claro e escuro. Seus dados implementam a interface de tabela do sgcUICore, IsgcUITable, de modo que um gráfico do **sgcCharts** pode plotar as linhas de uma grade com um TsgcChartDataSource_Table, e uma grade colocada em um bloco do **sgcDashboard** acompanha o tema do dashboard.



Edições

O sgcGrid é distribuído como Registered, Trial e Community. A edição Community não inclui a edição no local (SGC_GRID_EDIT) nem a ponte sgcGridWS com os feeds em tempo real e a consulta de IA. Consulte Edições.

Por onde começar

- Arquitetura: o modelo neutro em relação ao framework, o renderizador no canvas do sgcUICore, os controles VCL e FMX e o modelo de threads.
- Plataformas e Desempenho: as versões do Delphi, o C++Builder, as plataformas e o orçamento de quadro.
- Início Rápido VCL, Início Rápido FMX e Início Rápido C++Builder: uma primeira grade em poucos minutos.
- Componentes: as grades, as classes do modelo, as fontes de dados e a consulta de IA.
- Atualizações em tempo real e threads, Ordenação, Filtragem, Seleção e teclado e Redimensionamento de colunas.
- Flash ao alterar, Formatação condicional, Edição no local e Modo virtual.
- JSON e CSV, Vinculação de DataSet, Feeds em tempo real, Pergunte à sua grade e Gráficos a partir da grade.
- Temas e Renderização headless.
- Demos: os projetos de demonstração em Delphi e C++Builder.

Arquitetura

O sgcGrid é construído em camadas. Cada camada depende apenas das que estão abaixo dela, portanto o modelo, o renderizador e as units de importação e exportação compilam sem VCL nem FireMonkey e também são testados no Linux64.

Camada	Unit	Função
Modelo	sgcGrid_Columns, sgcGrid_Data	TsgcGridData contém as colunas e as linhas: um Variant por coluna, um carimbo de alteração e uma direção de alteração por célula, indexados pela chave da linha. Ele ordena, filtra e suporta um modo virtual. É seguro para threads e implementa IsgcUITable.
Visão de linhas	sgcGrid_Lines, sgcGrid_Group, sgcGrid_Summary	TsgcGridLineView mapeia as linhas de exibição de uma grade (linhas de grupo e linhas da visão) para as linhas da visão dos dados, TsgcGridGrouper constrói os grupos e sgcGrid_Summary calcula os resumos de grupo e de rodapé. Na edição Registered, sgcGrid_Tree e sgcGrid_Detail acrescentam as linhas de árvore e as faixas de detalhe.
Formatação	sgcGrid_Format	O flash ao alterar e as regras de formatação condicional, compartilhados por todos os renderizadores.
Renderizador	sgcGrid_Render	TsgcGridRenderer pinta um TsgcGridData em qualquer canvas do sgcUICore: cabeçalho, linhas, linhas de grade, seleção, glifos de ordenação, marcas de seleção e os estilos de célula. Somente as linhas visíveis são pintadas. Ele também grava arquivos PNG e SVG sem janela.
Importação e exportação	sgcGrid_IO, sgcGrid_Source_DataSet	Carga e gravação de JSON e CSV, e o TsgcGridSource_DataSet, que preenche uma grade a partir de qualquer TDataSet.
Exportação para Excel	sgcGrid_Zip, sgcGrid_XLSX	O gravador XLSX por trás de SaveToXLSX, somente na edição Registered. Veja Exportação para Excel.
Controle VCL	sgcGrid_VCL	TsgcVCLGrid possui um TsgcGridData e um renderizador e acrescenta as barras de rolagem, o mouse, o teclado, a área de transferência e os editores no local.
Controle FMX	sgcGrid_FMX	TsgcFMXGrid é o controle FireMonkey com as mesmas propriedades, métodos e eventos, barras de rolagem desenhadas e arrastar por toque.
Ponte	sgcGridWS_*	As fontes de feed WebSocket, MQTT e SSE e o TsgcGridAIQuery. Eles precisam do sgcWebSockets.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme, sgcUICore_Table	O runtime compartilhado: o canvas vetorial e o rasterizador, a base de controles VCL e FMX com controle de cadência de quadros (MaxFPS), os tokens de tema e a interface IsgcUITable.

Modelo de threads

Todo leitor e escritor público de TsgcGridData usa uma seção crítica reentrante interna, de modo que uma thread de feed pode chamar UpdateRow, AppendRow, UpdateCell e DeleteRow diretamente, sem Synchronize. Os dados geram seu OnChange na thread de escrita depois que o bloqueio é liberado. A grade não pinta ali: ela marca seu conteúdo como sujo e a base de controles do sgcUICore agenda uma repintura na thread principal, no máximo **MaxFPS** vezes por segundo (60 por padrão). As alterações da exibição, como uma nova linha ou uma reordenação, são postadas uma única vez para a thread principal, onde o intervalo de rolagem, a posição de cauda e a linha selecionada, localizada novamente pela chave, são atualizados.
BeginUpdate e EndUpdate agrupam qualquer número de alterações em um único OnChange. As colunas são alteradas a partir da thread principal; enquanto threads de escrita estiverem em execução, envolva as alterações de colunas em Lock e Unlock. Consulte Atualizações em tempo real e threads.

Exibição e armazenamento

As linhas são armazenadas em ordem de inserção. Sem colunas de ordenação e sem filtro, a grade mostra o armazenamento como ele está. Caso contrário, uma exibição é reconstruída de forma preguiçosa na primeira leitura após uma alteração: o filtro e depois uma ordenação estável por várias colunas. Uma alteração em uma coluna que não está ordenada, sem filtro ativo, mantém a exibição atual, de modo que uma coluna de preço que muda o tempo todo não custa uma reordenação.

Visão de linhas

As grades desenham e rolam linhas de exibição, não linhas da visão. Cada grade tem um TsgcGridLineView sobre os seus dados. Sem coluna agrupada, a visão de linhas é a identidade, a linha i é a linha da visão i, e nada é construído. O agrupamento, e na edição Registered as linhas de árvore e as faixas de detalhe, são estado da visão de linhas: os dados, suas linhas da visão e IsgcUITable permanecem planos, de modo que gráficos, grades dinâmicas e a ponte veem as mesmas linhas de antes. A visão de linhas se reconstrói de forma preguiçosa quando o LayoutVersion dos dados se move, e só atualiza os resumos de grupo quando apenas valores mudaram. Veja Agrupamento.

Veja também

- Plataformas
- Desempenho
- TsgcGridData
- TsgcGridRenderer
- TsgcGridLineView

Edições

O sgcGrid é compilado a partir do mesmo código-fonte em todas as edições. A edição apenas muda quais recursos uma compilação inclui e quais units são distribuídas.

Edição	Inclui
Registered	Tudo: as grades VCL e FMX, o modelo, a ordenação, a filtragem, o agrupamento com o painel de agrupamento, os resumos de grupo e de rodapé, as colunas congeladas, as linhas de árvore, as linhas mestre-detalhe, a exportação para Excel, a seleção, o flash, as regras de formatação, a edição no local, o modo virtual, JSON e CSV, a vinculação de DataSet, os temas, a saída headless em PNG e SVG e a ponte sgcGridWS com as origens de feed WebSocket, MQTT e SSE e a consulta de IA.
Trial	Os mesmos recursos de Registered. Uma compilação feita com SGC_TRIAL mostra um aviso de que é uma versão de avaliação depois que a janela de lançamento indicada na compilação tiver passado. Nada mais muda.
Community	Tudo, exceto a edição no local, as linhas de árvore, as linhas mestre-detalhe, a exportação para Excel e a ponte sgcGridWS.

O que a Community deixa de fora

- A edição no local. A edição Community é compilada com SGC_GRID_EDIT indefinido, portanto StartEdit, CommitEdit, CancelEdit, EditorActive e os eventos OnBeforeEdit, OnValidate, OnEdited e OnEditError não existem, e ReadOnly False não tem editor. Consulte Edição no local.
- Linhas de árvore, linhas mestre-detalhe e exportação para Excel. A edição Community compila com SGC_GRID_ADVANCED não definido, de modo que as propriedades Tree e Detail, ExpandNode, CollapseNode, ExpandDetail, CollapseDetail, DetailExpanded, DetailData, SaveToXLSX, os eventos OnTreeExpandChange, OnDetailExpandChange, OnGetDetailData e OnDetailCellClick e o componente TsgcGridDataStore não existem. Veja Linhas de Árvore, Mestre Detalhe e Exportação para Excel.
- A ponte sgcGridWS: as units sgcGridWS_*.pas e seus pacotes não são distribuídos, portanto as fontes de feed e o TsgcGridAIQuery não estão disponíveis. Consulte Feeds em tempo real e Pergunte à sua grade.

O agrupamento, o painel de agrupamento, os resumos de grupo e de rodapé e as colunas congeladas fazem parte de todas as edições, Community incluída. Veja Agrupamento, Resumos e Colunas Congeladas.

A edição é uma escolha em tempo de compilação em sgcVer.inc (SGC_COMMUNITY, que também desativa SGC_TRIAL), não uma opção em tempo de execução. O código que usa a API de edição, os membros de árvore, detalhe e Excel ou a ponte não compila contra uma build Community, veja Solução de Problemas.

Instalação

O sgcGrid faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcGrid** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcGrid

- **sgcUICore**, a base compartilhada dos produtos de layout e do sgcCharts. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcGrid é selecionado.
- **sgcGrid for sgcWebSockets**, listado em **sgcWebSockets bridges**. Instala os pacotes sgcGridWS com as fontes de feed e o TsgcGridAIQuery (veja Feeds em Tempo Real). O instalador só o oferece quando o sgcWebSockets já está instalado na máquina e a sua licença inclui o sgcGrid. Caso contrário, ele aparece esmaecido.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13`.
`Demos\Grid`	Os demos do sgcGrid. Veja Demos.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcGrid.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcGrid, junto com sgcCharts, sgcDashboard, sgcNavigation, sgcToolbar e sgcWizard. As pontes sgcWebSockets não fazem parte do instalador Community, e o sgcPDF e o sgcMaps não fazem parte do Community. Veja Edições para saber o que a compilação Community deixa de fora.

Instalações autônomas anteriores

Se o instalador encontrar uma instalação autônoma mais antiga do sgcUICore, como a gravada pelo instalador autônomo do sgcCharts, ele lista os library paths e os pacotes de design dessa instalação na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcGridD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcGrid<V>` / `dclsgcGrid<V>`	Os componentes do sgcGrid (runtime / design time).
`sgcGridWS<V>` / `dclsgcGridWS<V>`	A ponte sgcWebSockets (runtime / design time). Requer o sgcWebSockets.

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcGrid`, depois `dclsgcGrid`, e os pacotes da ponte por último. Em seguida, instale os pacotes de design time `dcl*` e adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcGridC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

.NET

Nada muda para o .NET. O sgcGrid continua sendo distribuído como seus próprios pacotes NuGet (`esegece.sgcGrid` e os pacotes complementares). Veja a instalação do .NET.

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13, com pacotes de runtime independentes de framework, VCL e FireMonkey e pacotes de tempo de projeto VCL e FireMonkey, veja Pacotes abaixo.
C++Builder	C++Builder 2007 ao C++Builder 13, com os cabeçalhos .hpp gerados e as bibliotecas estáticas. Consulte Início Rápido C++Builder.
VCL	TsgcVCLGrid no Windows, Win32 e, nas versões que o têm, Win64.
FireMonkey	TsgcFMXGrid a partir do Delphi XE6 (SGC_GRID_FMX), no Windows, macOS, iOS e Android.
Linux64	As units neutras em relação ao framework (o modelo, a formatação, o renderizador e a importação e exportação de JSON e CSV) a partir do Delphi 10.2 (SGC_GRID_LINUX), por exemplo para renderizar imagens de grade em um servidor.
Ponte	Os pacotes sgcGridWS exigem o pacote sgcGrid e o pacote sgcWebSockets da mesma versão do Delphi ou do C++Builder.
.NET	Uma porta em C# com a mesma API para WinForms, WPF e Avalonia, a saída headless em PNG e SVG, a fonte DataTable e as pontes com o sgcWebSockets e o sgcHTML, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão Geral do .NET.

Pacotes

O sgcGrid traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcGridD13.dpk para o Delphi 13 e sgcGridC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcGrid	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey: o modelo, as colunas, a formatação, o renderizador, o agrupamento, os resumos, as linhas de árvore e de detalhe, a importação e exportação de JSON, CSV e XLSX, TsgcGridDataStore e TsgcGridSource_DataSet.	rtl, dbrtl, sgcUICore
sgcGridVCL	Runtime, VCL	sgcGrid_VCL (TsgcVCLGrid)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcGrid
sgcGridFMX	Runtime, FireMonkey, a partir do XE6	sgcGrid_FMX (TsgcFMXGrid)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcGrid
dclsgcGrid	Tempo de projeto, VCL	Registra TsgcGridSource_DataSet, TsgcGridDataStore (não na edição Community) e TsgcVCLGrid na página **SGC Grid** da paleta, e um editor de propriedade para a Source das opções	designide, rtl, vcl, sgcGrid, sgcGridVCL

		de detalhe, que lista apenas os componentes que contêm dados da grade.	
dclsgcGridFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXGrid na página **SGC Grid** da paleta.	designide, rtl, fmx, sgcGrid, sgcGridFMX, dclsgcGrid
sgcGridWS, dclsgcGridWS	Runtime e tempo de projeto, ponte do sgcWebSockets	As fontes de feed em tempo real (TsgcGridFeedSource_WebSocket, TsgcGridFeedSource_MQTT, TsgcGridFeedSource_SSE) e TsgcGridAIQuery, independentes de framework, na página **SGC Grid** da paleta.	rtl, sgcGrid, sgcWebSockets

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcGridFMX exige o dclsgcGrid, portanto instale primeiro o dclsgcGrid. O instalador instala o sgcUICore junto com o sgcGrid. Se você compilar os pacotes por conta própria, compile primeiro o sgcUICore.

Versões do IDE e plataformas de destino

Delphi	sgcGrid	sgcGridVCL	sgcGridFMX	dclsgcGrid, dclsgcGridFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE. O sgcGridWS é compilado para as mesmas plataformas do sgcGrid, e o dclsgcGridWS para as mesmas plataformas do dclsgcGrid.

C++Builder	sgcGrid, sgcGridVCL	sgcGridFMX	dclsgcGrid, dclsgcGridFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcGrid)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcGrid)

XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcGridC13.cbproj, sgcGridVCLC13.cbproj, sgcGridFMXC13.cbproj, dclsgcGridC13.cbproj e dclsgcGridFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. O sgcGridWS e o dclsgcGridWS são compilados para as mesmas versões do C++Builder e plataformas que o sgcGrid e o dclsgcGrid, exceto o Win64x, porque precisam de uma compilação Win64x do pacote sgcWebSockets. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXGrid) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcGridVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcGrid_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcGrid é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa o modelo, a formatação, o renderizador e a importação e exportação de JSON e CSV, por exemplo para renderizar imagens da grade. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Veja também

- Edições
- Arquitetura

Desempenho

A grade pinta somente as linhas e colunas visíveis, obtém o bloqueio dos dados uma vez por quadro, mantém em cache a geometria das colunas e desenha o texto das células a partir de um cache de imagens de glifos no canvas raster. O custo de um quadro depende do tamanho da janela, não do número de linhas.

Orçamento de quadro

A suíte de QA tem um teste de desempenho: um quadro de 1920 x 1080 com 12 colunas, 100.000 linhas e 45 linhas visíveis, renderizado 60 vezes no canvas raster. O teste falha acima de 16 ms por quadro no Win64 e de 25 ms no Win32. As últimas médias medidas foram:

Plataforma	Quadro simples	Flash e 3 regras de formatação
Win32	3.0 ms	3.2 ms
Win64	4.6 ms	6.8 ms
Linux64	3.7 ms	

Assim, uma grade em 1080p é renderizada em poucos milissegundos, bem dentro dos 16,7 ms de um quadro a 60 fps.

Taxa de atualização

As atualizações e os quadros são desacoplados. Os escritores podem chamar UpdateRow com a frequência com que o feed entrega dados, e a grade se repinta no máximo MaxFPS vezes por segundo, independentemente de quantas linhas mudaram nesse intervalo. O demo LiveTicker grava cerca de 1.000 atualizações de preço por segundo, a partir de uma thread de trabalho, em uma grade ordenada e com flash a 60 fps.

Dicas

- Envolva um lote de alterações em BeginUpdate e EndUpdate para que a grade receba uma única notificação.
- Ordenar por uma coluna que muda o tempo todo reordena a exibição a cada alteração dessa coluna. As alterações em colunas que não estão ordenadas, sem filtro, mantêm a exibição atual.
- Use MaxRows para logs e feeds que crescem sem fim, e o modo virtual para dados muito grandes que já vivem em outro lugar.
- Reduza MaxFPS quando a grade só precisa parecer em tempo real, não fluida.

Veja também

- Atualizações em tempo real e threads
- Modo virtual
- Arquitetura

Início Rápido VCL

Em tempo de design

- Coloque um **TsgcVCLGrid** da página de paleta sgcGrid em um formulário e defina Align como alClient.
- Abra o editor de **Columns** e adicione uma coluna por campo. Defina FieldName, Caption e DataType, e DisplayFormat para números e datas.
- Preencha as linhas a partir do código com Data.AppendRow e mantenha-as atualizadas com Data.UpdateRow, a partir de qualquer thread.

A partir do 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 recebe os valores na ordem das colunas; UpdateRow recebe nomes de campos, portanto uma mensagem que traz apenas um preço atualiza apenas o preço. Uma chave que ainda não existe também é inserida por UpdateRow, a menos que Data.AutoInsert seja False.

Veja também

- Início Rápido FMX
- Atualizações em tempo real e threads
- TsgcVCLGrid

Início Rápido FMX

TsgcFMXGrid precisa do Delphi XE6 ou posterior. Coloque-o em um formulário FireMonkey ou crie-o a partir do código. Ele tem as mesmas colunas, dados e eventos da grade 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;
```

O FireMonkey não tem barras de rolagem nativas em um controle raster, portanto a grade desenha as suas. A roda do mouse rola três linhas, Shift rola na horizontal, e um gesto de arrastar rola com o dedo em telas sensíveis ao toque.

Veja também

- Início Rápido VCL
- TsgcFMXGrid

Início Rápido C++Builder

O C++Builder usa os mesmos componentes por meio dos cabeçalhos e das bibliotecas estáticas gerados pelos pacotes do C++Builder. Adicione `sgcUI\Source` e `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits) aos caminhos de include e de biblioteca, o que o instalador já faz para a IDE. Os cabeçalhos e as bibliotecas do sgcGrid e do sgcUICore estão nessa única pasta. Inclua os cabeçalhos sgcGrid_*.hpp e vincule sgcGridC13.lib e sgcUICoreC13.lib, ou instale os pacotes de design-time e coloque um TsgcVCLGrid em um formulário.
Os arrays abertos do Delphi são passados com a macro OPENARRAY. As chamadas a AppendRow e UpdateRow também são seguras para threads a partir do C++.

Veja também

- Início Rápido VCL
- Demos
- Solução de Problemas

Demos

Os demos estão na pasta Demos da instalação: os projetos Delphi em Demos\Delphi e os projetos C++Builder em Demos\CBuilder. Cada pasta de demo tem um readme.html. Cada demo também aceita duas opções usadas na verificação da compilação: /test executa suas verificações e termina com o número de falhas, e /snapshot salva a janela em um arquivo PNG.

Demo	Delphi	C++Builder	Mostra
LiveTicker	Sim	Sim	Cerca de 1.000 atualizações de preço por segundo a partir de uma thread de trabalho com UpdateRow, uma exibição ordenada por preço, alterações com flash, uma regra de formatação para alterações negativas, alternadores de TailMode e de tema escuro, e as atualizações e os quadros por segundo.
LiveTickerFMX	Sim	Não	O demo LiveTicker com um TsgcFMXGrid.
DataSetGrid	Sim	Sim	Um TsgcGridSource_DataSet vinculado a um TClientDataSet de 200 registros com KeyField, AutoColumns e LiveUpdates, uma seleção que acompanha o registro atual nos dois sentidos, e exportação e importação de JSON e CSV.
Editing	Sim	Sim	Edição no local com uma coluna de lista de opções, uma coluna booleana, OnValidate, OnEditError e OnEdited. Não está na edição Community.
GridAndChart	Sim	Sim	Um dashboard com um bloco de grade e um bloco de gráfico; um TsgcChartDataSource_Table vinculado à grade plota suas linhas. Precisa do sgcCharts e do sgcDashboard.
AskYourGrid	Sim	Sim	TsgcGridAIQuery filtrando, ordenando e destacando 200 ações a partir de perguntas em linguagem natural, com o analisador baseado em regras. Precisa do sgcWebSockets.
LiveFeed	Sim	Não	Um TsgcGridFeedSource_WebSocket alimentado por um servidor WebSocket no mesmo processo. Precisa do sgcWebSockets.

Observações sobre o C++Builder

- Os demos em C++Builder são compilados com o compilador clássico do C++Builder 13 para Win32 e Win64 e vinculam sgcGridC13.lib e sgcUICoreC13.lib.
- DataSetGrid vincula o MIDAS estaticamente no Win32. O C++Builder não tem biblioteca MIDAS estática para Win64, portanto a compilação de 64 bits precisa de midas.dll ao lado do executável.
- GridAndChart também vincula sgcChartsC13.lib e sgcDashboardC13.lib, e AskYourGrid vincula sgcGridWSC13.lib e sgcWebSocketsC13.lib.

Veja também

- Início Rápido VCL
- Início Rápido C++Builder
- Solução de Problemas

TsgcVCLGrid

A grade de dados em tempo real da VCL, na unit sgcGrid_VCL. Ela possui um TsgcGridData e um TsgcGridRenderer, pinta com o canvas raster do sgcUICore e implementa IsgcUITable (delegado a Data) e IsgcGridDataHost, de modo que uma fonte de dados de gráfico, uma fonte de feed ou uma fonte DataSet podem se vincular ao próprio componente de grade.
Ela usa as barras de rolagem nativas do Windows: a barra vertical conta linhas (linhas de exibição enquanto as linhas estão agrupadas) e a barra horizontal pixels lógicos, e a linha de cabeçalho permanece fixa. Os membros de edição existem apenas quando SGC_GRID_EDIT está definido.

Manipuladores de eventos dos exemplos de recursos

Os exemplos das páginas de Recursos atribuem aos eventos os métodos desta pequena classe.

```
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;
```

Propriedades

Propriedade	Tipo	Descrição
Columns	TsgcGridColumns	As colunas da grade. Consulte TsgcGridColumns.
Data	TsgcGridData	As linhas, a ordenação, o filtro e o rastreamento de alterações. Pública. Também implementa IsgcUITable para a grade. Consulte TsgcGridData.
Renderer	TsgcGridRenderer	O renderizador com o qual a grade desenha. Pública, somente leitura.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	O destaque da seleção e do flash de alteração. 0 usa o destaque do tema.
FontSize	Double	O tamanho do texto das células.

HeaderHeight, RowHeight	Integer	A altura do cabeçalho (32) e de cada linha (28), em pixels lógicos.
GridLinesH, GridLinesV	Boolean	Desenham as linhas de grade horizontais e verticais. True por padrão.
AlternateRows	Boolean	Escurece uma linha sim, outra não. True por padrão.
SelectionMode	TsgcGridSelectionMode	gsmRow (padrão) seleciona linhas inteiras, gsmCell seleciona células individuais.
BorderStyle	TBorderStyle (VCL), TsgcGridBorderStyle (FMX)	Uma moldura simples na cor Border do tema, ou nenhuma.
MaxRows	Integer	Modo de cauda: 0 mantém todas as linhas, acima de 0 descarta as linhas mais antigas ao inserir.
TailMode	Boolean	Mantém a linha mais nova visível enquanto a grade está rolada até o final.
FilterText	string	Mostra apenas as linhas cujo texto visível o contém, sem diferenciar maiúsculas de minúsculas.
ReadOnly	Boolean	True por padrão. False habilita a edição no local das colunas Editable (SGC_GRID_EDIT).
FlashChanges, FlashDuration, FlashMode	Boolean, Integer, TsgcGridFlashMode	Faz as células alteradas piscarem, por quantos milissegundos (800), no fundo ou no texto.
FlashUpColor, FlashDownColor, FlashChangedColor	TsgcUIColor	As cores do flash; 0 usa os padrões do tema.
FormatRules	TsgcGridFormatRules	As regras de formatação condicional. Consulte Formatação condicional.
MaxFPS	Integer	O maior número de repinturas por segundo (60).
FrameCount	Cardinal	O número de quadros pintados até agora. Pública, somente leitura.
TopRow, ScrollX	Integer	A primeira linha visível (a primeira linha de exibição enquanto as linhas estão agrupadas) e a posição de rolagem horizontal. Public.
SelectedRow, SelectedCol	Integer	A linha da exibição e o índice de coluna selecionados, -1 para nenhum. Pública.
SelectedKey	string	A chave da linha selecionada. A seleção acompanha a chave quando a exibição muda. Pública, somente leitura.
FrozenColumns	Integer	O número de colunas visíveis que não rolam na horizontal, 0 por padrão. Veja Colunas Congeladas.
ShowFooter	Boolean	Mostra a linha de rodapé com os resumos das colunas, False por padrão. Veja Resumos.
ShowGroupPanel, GroupPanelHeight	Boolean, Integer	Mostra o painel de agrupamento acima do cabeçalho (False por padrão) e sua altura em DIPs (36). Veja Agrupamento.

GroupIndent	Integer	O recuo de cada nível de grupo em DIPs, 20 por padrão.
GroupsExpanded	Boolean	O estado dos novos grupos, True por padrão. Defini-lo mantém o estado dos grupos atuais.
GroupSummaries	Boolean	Mostra os resumos das colunas nas linhas de grupo, True por padrão.
Lines	TsgcGridLineView	As linhas de exibição: linhas de grupo e linhas da visão. Public, somente leitura. Veja TsgcGridLineView.
FocusedLine	Integer	A linha de exibição com o foco, -1 para nenhuma. Em uma linha de grupo, SelectedRow é -1. Public.
Tree	TsgcGridTreeOptions	Linhas de árvore: KeyField, ParentField, TreeField, Indent e Expanded. O modo de árvore fica ativo quando KeyField e ParentField indicam colunas da grade. Somente na edição Registered. Veja Linhas de Árvore.
Detail	TsgcGridDetailOptions	Linhas mestre-detalhe: Active, Source, MasterKeyField, DetailKeyField, Lines e Columns. Somente na edição Registered. Veja Mestre Detalhe.

Métodos

Método	Retorna	Descrição
ScrollToRow(aRow)		Rola a linha da exibição até ficar visível.
RowAt(X, Y), ColAt(X, Y)	Integer	A linha da exibição e o índice de coluna sob um ponto, -1 para nenhum.
SaveToPNG(aFileName, aWidth, aHeight)		Grava a exibição atual em um arquivo PNG.
SaveToSVG(aFileName, aWidth, aHeight)		Grava a exibição atual em um arquivo SVG.
StartEdit(aRow, aCol)	Boolean	Abre o editor em uma célula, ou alterna uma célula booleana (SGC_GRID_EDIT).
CommitEdit	Boolean	Converte, valida e grava o texto do editor; False mantém o editor aberto (SGC_GRID_EDIT).
CancelEdit		Fecha o editor sem gravar (SGC_GRID_EDIT).
EditorActive, Editor, EditRow, EditCol		O estado do editor aberto (SGC_GRID_EDIT). Propriedades.
GroupBy(aFieldNames), ClearGrouping		Agrupa pelos campos indicados (FieldName, nível 0 primeiro; campos desconhecidos são ignorados), ou remove o agrupamento.
ExpandAll, CollapseAll		Expande ou recolhe todos os grupos, e os novos grupos daqui em diante. No modo de árvore, atuam sobre os nós da árvore (edição Registered).
ExpandGroup(aGroup), CollapseGroup(aGroup)		Expande ou recolhe um grupo.

GroupCount	Integer	O número de grupos.
GroupInfo(aGroup)	TsgcGridGroupInfo	Nível, pai, coluna, membro, contagem de linhas, estado e linha de um grupo.
FooterSummary(aCol), FooterSummaryText(aCol)	Variant, string	O valor de rodapé de uma coluna (índice de Columns) e o seu texto; Null e uma string vazia quando a coluna não tem SummaryType.
ExpandNode(aRow), CollapseNode(aRow)		Expande ou recolhe o nó da árvore da linha da visão aRow e dispara OnTreeExpandChange. Somente na edição Registered.
ExpandDetail(aRow), CollapseDetail(aRow)		Abre ou fecha a faixa de detalhe da linha mestre da visão aRow e dispara OnDetailExpandChange. Somente na edição Registered.
DetailExpanded(aRow)	Boolean	True quando a faixa da linha da visão aRow está aberta. Somente na edição Registered.
DetailData(aRow)	TsgcGridData	As linhas da faixa da linha da visão aRow, nil quando ela está recolhida. Somente leitura. Somente na edição Registered.
SaveToXLSX(aStream; aSheetName = 'Sheet1'), SaveToXLSX(aFileName; aSheetName = 'Sheet1')		Grava a grade como uma pasta de trabalho do Excel com as larguras de coluna atuais e FrozenColumns. Somente na edição Registered. Veja Exportação para Excel.

Eventos

Evento	Tipo	Descrição
OnCellClick, OnCellDblClick	TsgcGridCellEvent	Uma célula recebeu um clique ou um clique duplo: linha da exibição e índice de coluna.
OnSelectionChange	TNotifyEvent	A linha ou a célula selecionada mudou.
OnSortChange	TNotifyEvent	Um clique no cabeçalho mudou a ordenação.
OnColumnResized	TsgcGridColumnEvent	Uma coluna foi redimensionada com o mouse.
OnGetCellStyle	TsgcGridCellStyleEvent	A palavra final sobre as cores e o estilo de negrito de cada célula pintada. Consulte Formatação condicional.
OnBeforeEdit	TsgcGridBeforeEditEvent	Um editor está prestes a abrir; defina aAllow como False para recusar (SGC_GRID_EDIT).
OnValidate	TsgcGridValidateEvent	O valor convertido antes de ser gravado; altere-o ou rejeite-o (SGC_GRID_EDIT).
OnEdited	TsgcGridEditedEvent	Um valor foi gravado, com o valor antigo e o novo (SGC_GRID_EDIT).
OnEditError	TsgcGridEditErrorEvent	O texto não pôde ser convertido para o DataType da coluna; aMessage pode ser alterado (SGC_GRID_EDIT).

OnGrouping-Change	TNotifyEvent	As colunas agrupadas ou seus níveis mudaram, pelo painel de agrupamento ou por código.
OnGroupEx-pandChange	TsgcGridGrou-pExpandEvent	Um grupo foi expandido ou recolhido; aGroup é -1 depois de Ex-pandAll e CollapseAll.
OnGetGroupText	TsgcGridGet-GroupTextEvent	Altera o título aText da linha de grupo aGroup. Disparado com o bloqueio dos dados mantido.
OnTreeExpand-Change	TsgcGri-dRowExpandE-vent	Um nó da árvore foi expandido ou recolhido; aRow é -1 depois de ExpandAll e CollapseAll no modo de árvore. Somente na edição Registered.
OnDetailExpand-Change	TsgcGri-dRowExpandE-vent	Uma faixa de detalhe foi aberta ou fechada; aRow é a linha mestre da visão. Somente na edição Registered.
OnGetDetailData	TsgcGridGetDe-tailDataEvent	Preenche as linhas de uma faixa (aData) para a chave mestre aMasterKey; defina aHandled como True para mantê-las. Sender é a grade. Disparado sem nenhum bloqueio mantido. Somente na edição Registered.
OnDetailCellClick	TsgcGridDetail-CellEvent	Uma célula de uma faixa foi clicada: aRow é a linha mestre da visão, aDetailRow e aDetailCol a linha da visão e o índice de Columns dos dados da faixa. Somente na edição Registered.

Exemplo

```
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;
```

Veja também

- TsgcGridColumns
- TsgcGridData
- TsgcGridRenderer
- Seleção e teclado
- Edição no local
- Agrupamento
- Resumos

- Colunas Congeladas
- TsgcGridLineView
- Linhas de Árvore
- Mestre Detalhe
- Exportação para Excel

TsgcFMXGrid

A grade de dados em tempo real do FireMonkey, na unit sgcGrid_FMX (Delphi XE6 e posteriores, Windows, macOS, iOS e Android). Ela publica as propriedades, os métodos e os eventos de TsgcVCLGrid, além dos membros usuais do FireMonkey, como Position, Size, Margins, Opacity e os eventos de mouse e de teclado.
A grade desenha as próprias barras de rolagem nas bordas direita e inferior enquanto o conteúdo for maior que a área visível: o polegar pode ser arrastado e um clique na trilha rola por páginas. A roda do mouse rola três linhas (Shift: na horizontal) e um gesto de arrastar rola com o dedo. Ctrl+C copia a seleção, Cmd+C no macOS. BorderStyle é um TsgcGridBorderStyle, gbsNone ou gbsSingle. A partir de uma thread de escrita, a grade enfileira uma única atualização da exibição para a thread principal com TThread.Queue.

Propriedades

Propriedade	Tipo	Descrição
Columns	TsgcGridColumns	As colunas da grade. Consulte TsgcGridColumns.
Data	TsgcGridData	As linhas, a ordenação, o filtro e o rastreamento de alterações. Pública. Também implementa IsgcUITable para a grade. Consulte TsgcGridData.
Renderer	TsgcGridRenderer	O renderizador com o qual a grade desenha. Pública, somente leitura.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	O destaque da seleção e do flash de alteração. 0 usa o destaque do tema.
FontSize	Double	O tamanho do texto das células.
HeaderHeight, RowHeight	Integer	A altura do cabeçalho (32) e de cada linha (28), em pixels lógicos.
GridLinesH, GridLinesV	Boolean	Desenham as linhas de grade horizontais e verticais. True por padrão.
AlternateRows	Boolean	Escurece uma linha sim, outra não. True por padrão.
SelectionMode	TsgcGridSelectionMode	gsmRow (padrão) seleciona linhas inteiras, gsmCell seleciona células individuais.
BorderStyle	TBorderStyle (VCL), TsgcGridBorderStyle (FMX)	Uma moldura simples na cor Border do tema, ou nenhuma.
MaxRows	Integer	Modo de cauda: 0 mantém todas as linhas, acima de 0 descarta as linhas mais antigas ao inserir.
TailMode	Boolean	Mantém a linha mais nova visível enquanto a grade está rolada até o final.
FilterText	string	Mostra apenas as linhas cujo texto visível o contém, sem diferenciar maiúsculas de minúsculas.
ReadOnly	Boolean	True por padrão. False habilita a edição no local das colunas Editable (SGC_GRID_EDIT).

FlashChanges, FlashDuration, FlashMode	Boolean, Integer, TsgcGridFlashMode	Faz as células alteradas piscarem, por quantos milissegundos (800), no fundo ou no texto.
FlashUpColor, FlashDownColor, FlashChangedColor	TsgcUIColor	As cores do flash; 0 usa os padrões do tema.
FormatRules	TsgcGridFormatRules	As regras de formatação condicional. Consulte Formatação condicional.
MaxFPS	Integer	O maior número de repinturas por segundo (60).
FrameCount	Cardinal	O número de quadros pintados até agora. Pública, somente leitura.
TopRow, ScrollX	Integer	A primeira linha visível (a primeira linha de exibição enquanto as linhas estão agrupadas) e a posição de rolagem horizontal. Public.
SelectedRow, SelectedCol	Integer	A linha da exibição e o índice de coluna selecionados, -1 para nenhum. Pública.
SelectedKey	string	A chave da linha selecionada. A seleção acompanha a chave quando a exibição muda. Pública, somente leitura.
FrozenColumns	Integer	O número de colunas visíveis que não rolam na horizontal, 0 por padrão. Veja Colunas Congeladas.
ShowFooter	Boolean	Mostra a linha de rodapé com os resumos das colunas, False por padrão. Veja Resumos.
ShowGroupPanel, GroupPanelHeight	Boolean, Integer	Mostra o painel de agrupamento acima do cabeçalho (False por padrão) e sua altura em DIPs (36). Veja Agrupamento.
GroupIndent	Integer	O recuo de cada nível de grupo em DIPs, 20 por padrão.
GroupsExpanded	Boolean	O estado dos novos grupos, True por padrão. Defini-lo mantém o estado dos grupos atuais.
GroupSummaries	Boolean	Mostra os resumos das colunas nas linhas de grupo, True por padrão.
Lines	TsgcGridLineView	As linhas de exibição: linhas de grupo e linhas da visão. Public, somente leitura. Veja TsgcGridLineView.
FocusedLine	Integer	A linha de exibição com o foco, -1 para nenhuma. Em uma linha de grupo, SelectedRow é -1. Public.
Tree	TsgcGridTreeOptions	Linhas de árvore: KeyField, ParentField, TreeField, Indent e Expanded. O modo de árvore fica ativo quando KeyField e ParentField indicam colunas da grade. Somente na edição Registered. Veja Linhas de Árvore.
Detail	TsgcGridDetailOptions	Linhas mestre-detalhe: Active, Source, MasterKeyField, DetailKeyField, Lines e Columns. Somente na edição Registered. Veja Mestre Detalhe.

Métodos

Método	Retorna	Descrição
ScrollToRow(aRow)		Rola a linha da exibição até ficar visível.
RowAt(X, Y), ColAt(X, Y)	Integer	A linha da exibição e o índice de coluna sob um ponto, -1 para nenhum.
SaveToPNG(aFileName, aWidth, aHeight)		Grava a exibição atual em um arquivo PNG.
SaveToSVG(aFileName, aWidth, aHeight)		Grava a exibição atual em um arquivo SVG.
StartEdit(aRow, aCol)	Boolean	Abre o editor em uma célula, ou alterna uma célula booleana (SGC_GRID_EDIT).
CommitEdit	Boolean	Converte, valida e grava o texto do editor; False mantém o editor aberto (SGC_GRID_EDIT).
CancelEdit		Fecha o editor sem gravar (SGC_GRID_EDIT).
EditorActive, Editor, EditRow, EditCol		O estado do editor aberto (SGC_GRID_EDIT). Propriedades.
GroupBy(aFieldNames), ClearGrouping		Agrupa pelos campos indicados (FieldName, nível 0 primeiro; campos desconhecidos são ignorados), ou remove o agrupamento.
ExpandAll, CollapseAll		Expande ou recolhe todos os grupos, e os novos grupos daqui em diante. No modo de árvore, atuam sobre os nós da árvore (edição Registered).
ExpandGroup(aGroup), CollapseGroup(aGroup)		Expande ou recolhe um grupo.
GroupCount	Integer	O número de grupos.
GroupInfo(aGroup)	TsgcGridGroupInfo	Nível, pai, coluna, membro, contagem de linhas, estado e linha de um grupo.
FooterSummary(aCol), FooterSummaryText(aCol)	Variant, string	O valor de rodapé de uma coluna (índice de Columns) e o seu texto; Null e uma string vazia quando a coluna não tem SummaryType.
ExpandNode(aRow), CollapseNode(aRow)		Expande ou recolhe o nó da árvore da linha da visão aRow e dispara OnTreeExpandChange. Somente na edição Registered.
ExpandDetail(aRow), CollapseDetail(aRow)		Abre ou fecha a faixa de detalhe da linha mestre da visão aRow e dispara OnDetailExpandChange. Somente na edição Registered.
DetailExpanded(aRow)	Boolean	True quando a faixa da linha da visão aRow está aberta. Somente na edição Registered.

DetailData(aRow)	TsgcGridData	As linhas da faixa da linha da visão aRow, nil quando ela está recolhida. Somente leitura. Somente na edição Registered.
SaveToXLSX(aStream; aSheetName = 'Sheet1'), SaveToXLSX(aFileName; aSheetName = 'Sheet1')		Grava a grade como uma pasta de trabalho do Excel com as larguras de coluna atuais e FrozenColumns. Somente na edição Registered. Veja Exportação para Excel.

Eventos

Evento	Tipo	Descrição
OnCellClick, OnCellDblClick	TsgcGridCellEvent	Uma célula recebeu um clique ou um clique duplo: linha da exibição e índice de coluna.
OnSelectionChange	TNotifyEvent	A linha ou a célula selecionada mudou.
OnSortChange	TNotifyEvent	Um clique no cabeçalho mudou a ordenação.
OnColumnResized	TsgcGridColumnEvent	Uma coluna foi redimensionada com o mouse.
OnGetCellStyle	TsgcGridCellStyleEvent	A palavra final sobre as cores e o estilo de negrito de cada célula pintada. Consulte Formatação condicional.
OnBeforeEdit	TsgcGridBeforeEditEvent	Um editor está prestes a abrir; defina aAllow como False para recusar (SGC_GRID_EDIT).
OnValidate	TsgcGridValidateEvent	O valor convertido antes de ser gravado; altere-o ou rejeite-o (SGC_GRID_EDIT).
OnEdited	TsgcGridEditedEvent	Um valor foi gravado, com o valor antigo e o novo (SGC_GRID_EDIT).
OnEditError	TsgcGridEditErrorEvent	O texto não pôde ser convertido para o DataType da coluna; aMessage pode ser alterado (SGC_GRID_EDIT).
OnGroupingChange	TNotifyEvent	As colunas agrupadas ou seus níveis mudaram, pelo painel de agrupamento ou por código.
OnGroupExpandChange	TsgcFMXGridGroupExpandEvent	Um grupo foi expandido ou recolhido; aGroup é -1 depois de ExpandAll e CollapseAll.
OnGetGroupText	TsgcGridGetGroupTextEvent	Altera o título aText da linha de grupo aGroup. Disparado com o bloqueio dos dados mantido.
OnTreeExpandChange	TsgcFMXGridRowExpandEvent	Um nó da árvore foi expandido ou recolhido; aRow é -1 depois de ExpandAll e CollapseAll no modo de árvore. Somente na edição Registered.
OnDetailExpandChange	TsgcFMXGridRowExpandEvent	Uma faixa de detalhe foi aberta ou fechada; aRow é a linha mestre da visão. Somente na edição Registered.
OnGetDetailData	TsgcGridGetDetailDataEvent	Preenche as linhas de uma faixa (aData) para a chave mestre aMasterKey; defina aHandled como True para mantê-las. Sender é

		a grade. Disparado sem nenhum bloqueio mantido. Somente na edição Registered.
OnDetailCell-Click	TsgcFMXGridDe-tailCellEvent	Uma célula de uma faixa foi clicada: aRow é a linha mestre da visão, aDetailRow e aDetailCol a linha da visão e o índice de Columns dos dados da faixa. Somente na edição Registered.

Exemplo

```
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;
```

Veja também

- TsgcGridColumns
- TsgcGridData
- TsgcGridRenderer
- Seleção e teclado
- Edição no local
- Agrupamento
- Resumos
- Colunas Congeladas
- TsgcGridLineView
- Linhas de Árvore
- Mestre Detalhe
- Exportação para Excel

TsgcGridColumn

Uma coluna de uma grade, um item de TsgcGridColumns, na unit sgcGrid_Columns. Os dados endereçam os valores pelo ID da coluna, de modo que mover ou excluir uma coluna nunca desloca os valores das demais.

Propriedades

Propriedade	Tipo	Descrição
FieldName	string	O nome que os dados, os feeds, as importações e as regras de formatação usam para a coluna.
Caption	string	O texto do cabeçalho.
DataType	TsgcGridDataType	gdtText (padrão), gdtInteger, gdtFloat, gdtCurrency, gdtDateTime ou gdtBoolean. Ele orienta a ordenação, a formatação, o alinhamento, a edição e as comparações das regras.
Width, MinWidth	Integer	A largura (100) e a menor largura que um redimensionamento com o mouse permite (20), em pixels lógicos.
Alignment	TsgcGridAlignment	gaLeft, gaCenter, gaRight ou gaAuto (padrão): números à direita, texto à esquerda, booleanos centralizados.
DisplayFormat	string	Um formato de FormatFloat para números ou um formato de FormatDateTime para datas. Uma coluna booleana sem DisplayFormat é desenhada como uma marca de seleção.
Visible	Boolean	Mostra a coluna. As colunas ocultas mantêm seus valores e seu índice.
Editable	Boolean	A coluna pode ser editada no local quando a grade não é ReadOnly. True por padrão.
SortOrder, SortIndex	TsgcGridSortOrder, Integer	gsoNone, gsoAscending ou gsoDescending, e a posição da coluna em uma ordenação por várias colunas.
PickList	TStrings	Os valores do editor de caixa de combinação da coluna.
Tag	Integer	Livre para a aplicação.
GroupIndex	Integer	0 ou mais agrupa a grade por esta coluna nesse nível (níveis ordenados por GroupIndex, depois por Index); -1 (padrão) não agrupa. Veja Agrupamento.
SummaryType	TsgcGridSummaryType	gsuNone (padrão), gsuSum, gsuCount, gsuAverage, gsuMin ou gsuMax: o resumo da coluna no rodapé e nas linhas de grupo. Veja Resumos.
SummaryFormat	string	O padrão de FormatFloat ou FormatDateTime do resumo, texto literal entre aspas duplas; vazio usa DisplayFormat, e 0 para gsuCount.

Métodos

Método	Retorna	Descrição
IsNumeric	Boolean	True para gdtInteger, gdtFloat e gdtCurrency.
EffectiveAlignment	TsgcGridAlignment	O alinhamento para o qual gaAuto é resolvido.

Exemplo

```
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;
```

Veja também

- TsgcGridColumns
- Ordenação
- Edição no local
- Agrupamento
- Resumos

TsgcGridColumns

A coleção de itens TsgcGridColumn de uma grade ou de um TsgcGridData independente, na unit sgcGrid_Columns.

Propriedades

Propriedade	Tipo	Descrição
Items[Index]	TsgcGridColumn	As colunas, propriedade padrão.
VisibleColumn[Index]	TsgcGridColumn	As colunas visíveis, em ordem.
Version	Cardinal	Incrementado a cada alteração das colunas.

Métodos

Método	Retorna	Descrição
Add	TsgcGridColumn	Adiciona uma coluna vazia.
Add(aFieldName, aCaption, aDataType)	TsgcGridColumn	Adiciona uma coluna com um nome de campo, um título e um tipo de dados (gdtText por padrão).
FindByField(aFieldName)	TsgcGridColumn	A coluna de um nome de campo, nil quando não existe.
IndexOfField(aFieldName)	Integer	O índice da coluna de um nome de campo, -1 quando não existe.
VisibleCount	Integer	O número de colunas visíveis.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Qualquer propriedade de uma coluna mudou.
OnModelChange	TNotifyEvent	Uma alteração que afeta os dados ou a exibição, como uma ordenação ou uma nova coluna.

Exemplo

```
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;
```

Veja também

- TsgcGridColumn
- TsgcGridData

TsgcGridData

O armazenamento de linhas de sgcGrid, neutro em relação ao framework e seguro para threads, na unit sgcGrid_Data. Cada grade possui um em sua propriedade Data, e ele pode ser usado sozinho, por exemplo com TsgcGridRenderer. Create(nil) cria e possui as próprias colunas.
As linhas são mantidas em ordem de inserção e localizadas por chave em O(log n). Os índices de linha de Cells, CellText, KeyOf, RowOfKey e CellChange são linhas da exibição (ordenadas e filtradas); os índices de coluna são índices de Columns, inclusive das colunas ocultas. Uma leitura fora do intervalo retorna Null, uma string vazia ou cdNone e nunca gera exceção, porque uma thread de escrita pode excluir uma linha entre uma contagem e uma leitura.

Propriedades

Propriedade	Tipo	Descrição
Columns	TsgcGridColumns	As colunas.
Cells[aRow, aCol]	Variant	O valor de uma linha da exibição.
CellText[aRow, aCol]	string	O valor formatado com o DisplayFormat da coluna.
StoreCells[aStoreRow, aCol]	Variant	O valor de uma linha do armazenamento, para OnFilterRow.
FormatSettings	TFormatSettings	As configurações usadas para formatar e analisar valores.
AutoInsert	Boolean	UpdateRow e UpdateCell inserem uma chave ausente. True por padrão.
MaxRows	Integer	0 ilimitado; caso contrário, as linhas mais antigas são descartadas ao inserir.
FilterText	string	Mantém as linhas cujo texto visível o contém, sem diferenciar maiúsculas de minúsculas.
Virtual, VirtualRowCount	Boolean, Integer	O modo virtual e seu número de linhas. Consulte Modo virtual.
LayoutVersion	Cardinal	Incrementado quando o agrupamento ou o layout de árvore das linhas da visão pode ter mudado: inserção, exclusão, Clear, qualquer mudança de visão ou de coluna, uma mudança das colunas agrupadas, e uma mudança de valor em uma coluna com GroupIndex 0 ou mais ou em um campo passado a SetLayoutFields. Uma mudança de valor em qualquer outra coluna o mantém. A visão de linhas das grades o lê. Somente leitura.

Métodos

Método	Retorna	Descrição
AppendRow(aKey, aValues)	Integer	Adiciona uma linha com os valores na ordem das colunas e retorna sua linha do armazenamento. Uma chave vazia gera

		uma chave única #<n>; uma chave existente atualiza essa linha.
UpdateRow(aKey, aFields, aValues)		Define os campos indicados de uma linha. Campos desconhecidos são ignorados.
UpdateCell(aKey, aField, aValue)		Define um campo de uma linha.
DeleteRow(aKey)	Boolean	Exclui a linha de uma chave.
Clear		Exclui todas as linhas.
BeginUpdate, EndUpdate, IsUpdating		Agrupam as alterações em um único OnChange.
Lock, Unlock		Mantêm o bloqueio interno durante várias leituras.
RowCount, ViewCount	Integer	As linhas armazenadas (VirtualRowCount no modo virtual) e as linhas que passam pelo filtro.
ViewRow(aRow)	Integer	A linha do armazenamento de uma linha da exibição, -1 quando fora do intervalo.
KeyOf(aRow), RowOfKey(aKey), StoreKeyOf(aStoreRow)		Convertem entre linhas da exibição, linhas do armazenamento e chaves. RowOfKey retorna -1 quando a chave não existe ou foi filtrada.
CellChange(aRow, aCol)	TsgcGridCellChange	A Direction (cdNone, cdUp, cdDown, cdChanged) e a Age em milissegundos da última alteração de uma célula.
GetTick	Int64	O relógio monotônico em milissegundos dos carimbos de alteração.
InvalidateView		Reconstrói a exibição, por exemplo depois que a lógica de OnFilterRow mudou.
UseInvariantFormat		Define FormatSettings com separador decimal ponto, separador de milhares vírgula, yyyy-mm-dd e hh:nn:ss.
SetLayoutFields(aFields)		Campos (FieldName, sem diferenciar maiúsculas de minúsculas) cujas mudanças de valor também movem LayoutVersion, como as colunas agrupadas; um array vazio limpa a lista.

Eventos

Evento	Tipo	Descrição
OnChange	TsgcGridChangeEvent	gckData, gckView ou gckStructure, gerado depois que o bloqueio é liberado na thread de escrita.
OnFilterRow	TsgcGridFilterRowEvent	Aceita ou rejeita uma linha do armazenamento. Chamado com o bloqueio mantido.
OnGetValue	TsgcGridGetValueEvent	Os valores do modo virtual.

OnGetTick	TsgcGridGetTic-kEvent	Substitui o relógio dos carimbos de alteração, por exemplo em testes.

TsgcGridData também implementa IsgcUITable, consulte IsgcGridDataHost e IsgcUITable.

Exemplo

```
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;
```

Veja também

- Atualizações em tempo real e threads
- Filtragem
- Ordenação
- TsgcGridLineView

TsgcGridLineView

A visão de linhas da sgcGrid, independente de framework, na unit sgcGrid_Lines. Ela mapeia as linhas de exibição de uma grade para as linhas da visão do seu TsgcGridData. Cada grade tem uma em sua propriedade Lines, e ela pode ser usada isoladamente com um TsgcGridData e o TsgcGridRenderer. Ela não é dona dos dados: libere-a antes dos dados.
Uma visão de linhas é estado da visão. Os dados, suas linhas da visão e IsgcUITable não mudam. Cada linha tem a altura de uma linha de dados, então um índice de linha é uma posição de rolagem. Sem coluna agrupada (e sem árvore nem faixa de detalhe expandida na edição Registered), ou com dados virtuais, nada é construído e a linha i é a linha da visão i: esta é a identidade, a grade plana.

Tipos de linha

TsgcGridLineKind	Linha
glkRow	Uma linha da visão. Row a retorna.
glkGroup	Uma linha de grupo. Group retorna o grupo, Row retorna -1.
glkDetail	Uma linha de uma faixa de detalhe (edição Registered, veja Mestre Detalhe). Row retorna a linha mestre.

As linhas agrupadas são as linhas de grupo em ordem de profundidade, cada uma seguida de seus grupos filhos, e um grupo do último nível pelas suas linhas da visão. As linhas de grupo têm Level 0 no nível mais externo, as linhas de dados têm Level = GroupLevels.

Métodos

Método	Retorna	Descrição
Create(aData)		Cria uma visão de linhas sobre aData.
Update		Atualiza as linhas com os dados. Todo leitor abaixo a chama, então ela não custa nada quando nada mudou.
Invalidate		Força uma reconstrução na próxima leitura.
Identity	Boolean	True quando a linha i é a linha da visão i.
Count	Integer	O número de linhas de exibição.
Kind(aLine)	TsgcGridLineKind	O tipo de uma linha, glkRow fora do intervalo.
Row(aLine)	Integer	A linha da visão de uma linha de dados, a linha mestre de uma linha de detalhe, -1 para uma linha de grupo ou fora do intervalo.
Level(aLine)	Integer	O nível de recuo de uma linha.
Group(aLine)	Integer	O grupo de uma linha de grupo, o grupo mais interno de uma linha de dados, -1 na identidade ou fora do intervalo.
Expandable(aLine), Expanded(aLine)	Boolean	Se a linha pode ser expandida ou recolhida, e o seu estado.

ToggleLine(aLine)	Boolean	Alterna a linha; False quando não há nada para alternar.
LineOfRow(aRow)	Integer	A linha de uma linha da visão, -1 dentro de um grupo recolhido ou fora do intervalo.
GroupLevels, GroupCount	Integer	O número de níveis agrupados e de grupos.
GroupInfo(aGroup)	TsgcGrid-GroupInfo	Um grupo, de `0` a `GroupCount - 1` em ordem de profundidade.
GroupSummary(aGroup, aCol)	Variant	O resumo de uma coluna (índice de Columns) sobre as linhas do grupo; Null quando a coluna não tem SummaryType ou aGroup está fora do intervalo.
GroupBy(aFieldNames), ClearGrouping		Definem GroupIndex nas colunas indicadas (nível 0 primeiro) e -1 nas outras, ou -1 em todas as colunas.
SetGroupExpanded(aGroup, aExpanded)		Expande ou recolhe um grupo.
ExpandAll, CollapseAll		Todos os grupos, e os novos grupos daqui em diante.
FlattenAll	TsgcGrid-FlatLines	Cada linha de grupo e cada linha da visão em ordem de linhas, independentemente do que esteja recolhido; Hidden marca as linhas dentro de um grupo recolhido. As linhas de detalhe não são listadas.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcGrid-Data	Os dados da visão de linhas. Somente leitura.
GroupsExpanded	Boolean	O estado dos grupos que não foram alternados e dos novos grupos. Defini-lo mantém o estado dos grupos atuais.

Os membros de árvore e detalhe (TreeMode, NodeExpanded, SetNodeExpanded, DetailExpandable, DetailExpanded, SetDetailExpanded, ToggleDetail, DetailView, DetailIndex, TreeOptions, DetailOptions, Tree e Detail) existem apenas na edição Registered, veja Linhas de Árvore e Mestre Detalhe.

TsgcGridGroupInfo

Campo	Tipo	Descrição
Level	Integer	0 para o nível mais externo.
Parent	Integer	O grupo pai, -1 no nível 0.
Column	Integer	O índice de Columns da coluna agrupada.
RowCount	Integer	As linhas da visão do grupo, incluindo seus grupos filhos.
FirstRow	Integer	A linha da visão da primeira linha do grupo, em ordem de linhas.
Text	string	O texto do membro, (blank) para o membro vazio.

Value	Variant	O valor tipado da primeira linha, Null para o membro vazio.
Blank, Expanded	Boolean	Se o grupo é o membro vazio, e o seu estado.
Line	Integer	A linha da linha de grupo, -1 dentro de um grupo recolhido.
Path	string	Os textos de membro a partir do nível 0, unidos por #1.

TsgcGridFlatLines é um registro com Count e os arrays Kind, Ref (a linha da visão ou o grupo), Level e Hidden.

Threads

Chame a visão de linhas a partir da thread que é dona da grade, a thread principal para um controle. As threads de escrita continuam gravando nos dados; a visão de linhas se reconstrói na próxima leitura quando o LayoutVersion dos dados mudou, e só atualiza os resumos de grupo quando apenas valores mudaram. TsgcGridGrouper (unit sgcGrid_Group) faz o agrupamento para ela.

Exemplo

Agrupa três linhas por Region sem um controle e imprime cada grupo com sua contagem de linhas e a soma de Amount, depois os produtos sob ele.

```
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;
```

Veja também

- Agrupamento
- Resumos
- TsgcGridData
- TsgcGridRenderer
- Agrupamento e Resumos do .NET

TsgcGridDataStore

Um componente não visual, na unit sgcGrid_Detail, que guarda linhas sem mostrá-las. Sua função é ser o Source das faixas de detalhe de uma grade, veja Mestre Detalhe. Ele é preenchido por um TsgcGridSource_DataSet cujo Grid é o store, ou por código, de modo que um detalhe de TDataSet funciona a partir do designer de formulários sem código. Ele está na página SGC Grid da paleta de componentes.
Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.
TsgcGridDataStore descende de TComponent e implementa IsgcGridDataHost e IsgcUITable, esta última delegada a Data.

Propriedades

Propriedade	Tipo	Descrição
Columns	TsgcGridColumns	As colunas das linhas. Publicadas, de modo que são definidas no designer. Com um TsgcGridSource_DataSet e AutoColumns True, são criadas a partir dos campos.
Data	TsgcGridData	As linhas, lidas e gravadas como o Data de uma grade: AppendRow, UpdateRow, DeleteRow, BeginUpdate e EndUpdate. Public, somente leitura.

Uso

- Defina Detail.Source de uma grade como o store, e DetailKeyField como a coluna do store que guarda a chave mestre.
- As alterações nas linhas do store, por código ou pela origem TDataSet, preenchem de novo as faixas de detalhe abertas no próximo quadro.
- Liberar o store define Detail.Source da grade como nil.

Exemplo

Um TDataSet de pedidos carregado em um store e mostrado como o detalhe de uma grade 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;
```

O mesmo a partir do designer: defina DataSet, Grid (o store), KeyField e Active no TsgcGridSource_DataSet, e depois Detail.Source, Detail.MasterKeyField, Detail.DetailKeyField e Detail.Active na grade. A demo TreeAndDetail funciona assim.

Veja também

- Mestre Detalhe
- TsgcGridData
- TsgcGridColumns
- TsgcGridSource_DataSet
- Edições
- Linhas de Árvore e Mestre Detalhe do .NET

TsgcGridRenderer

O renderizador de sgcGrid, neutro em relação ao framework, na unit sgcGrid_Render. Ele pinta um TsgcGridData em qualquer canvas do sgcUICore, raster ou SVG, de modo que a grade VCL, a grade FMX e a saída headless ficam idênticas. A geometria é em pixels lógicos; Render recebe o tamanho do canvas em pixels do dispositivo e a escala, e alinha cada borda a pixels inteiros do dispositivo.

TsgcGridRenderOptions

A aparência publicada de uma grade: HeaderHeight, RowHeight, CellPadding (8), GridLinesH, GridLinesV, AlternateRows, SelectionMode, Theme, AccentColor, FontSize, HeaderFontSize, ShowSortGlyphs, FlashChanges, FlashDuration, FlashUpColor, FlashDownColor, FlashChangedColor, FlashMode e FormatRules, e o layout da visão de linhas FrozenColumns, ShowFooter, ShowGroupPanel, GroupPanelHeight, GroupIndent e GroupSummaries. Os controles de grade publicam as mesmas propriedades. Colors retorna as cores resolvidas do tema.

TsgcGridRenderState

Um registro com o estado de visualização de um quadro: ScrollX, ScrollY, SelectedRow, SelectedCol, HotRow, HotCol, FocusRow, FocusCol, Focused, FocusLine, HotLine e ColWidths. sgcGridRenderState retorna um estado vazio sem nada selecionado. FocusLine e HotLine são linhas de exibição, -1 por padrão; FocusLine -1 dá o foco à linha de FocusRow.

Propriedades

Propriedade	Tipo	Descrição
Options	TsgcGridRenderOptions	A aparência.
CurrentData, CurrentNow	TsgcGridData, Int64	Os dados e o tick do quadro que está sendo pintado, válidos em OnGetCellStyle.
Fading	Boolean	True depois de um quadro que pintou um flash ainda em andamento.
FooterSummaries	TsgcGridSummaryCache	Os valores de rodapé, atualizados a cada quadro com ShowFooter.

Métodos

Método	Retorna	Descrição
Render(aData, aState, aCanvas, aWidth, aHeight, aScale, aNow)		Pinta um quadro.
SaveToPNG, SaveToSVG		Gravam um quadro em um stream ou em um arquivo, com uma escala opcional.
HitTest(aData, aState, aX, aY)	TsgcGridHitInfo	A parte (ghpNone, ghpHeader, ghpHeaderResize, ghpCell), a linha e a coluna sob um ponto.

ColumnWidth, ColumnLeft, ColumnRight, ContentWidth, ContentHeight, VisibleRowRange		Auxiliares de geometria.
Render(aData, aLines, aState, aCanvas, aWidth, aHeight, aScale, aNow)		Desenha um quadro de linhas de exibição: o painel de agrupamento, o cabeçalho, as linhas e o rodapé. aLines nil é a grade plana. Veja TsgcGridLineView.
SaveToPNG(aData, aLines, ...), SaveToSVG(aData, aLines, ...)		Gravam um quadro de linhas de exibição em um stream ou em um arquivo.
HitTest(aData, aLines, aState, aX, aY, aHeight)	TsgcGridHitInfo	Teste de acerto sobre linhas de exibição. Passe a altura da viewport em aHeight para acertar o rodapé; com 0 o rodapé nunca é acertado.
VisibleLineRange, GutterWidth, FrozenWidth, ContentWidth(aColumns, aLines, aState)		Geometria da visão de linhas: as linhas visíveis sem o painel de agrupamento e o rodapé, a largura dos chevrons de grupo, a largura que não rola e a largura total.
GroupPanelDropIndex, GroupChipBounds, PaintDragChip		O ponto de soltura no painel de agrupamento, os limites de um chip e o desenho do chip arrastado, para hosts que arrastam cabeçalhos até o painel de agrupamento.
GroupText(aLines, aGroup)	string	O título de uma linha de grupo, com OnGetGroupText aplicado.

Sobre linhas, TsgcGridHitPart acrescenta ghpGroupPanel, ghpGroupChip, ghpGroupRow, ghpExpandButton (o chevron), ghpFooter e ghpDetail, e TsgcGridHitInfo acrescenta Line (a linha de exibição, -1 fora das linhas), Group (o grupo da linha, o chip em ghpGroupChip, o ponto de soltura em ghpGroupPanel), DetailRow e DetailCol.

Eventos

Evento	Tipo	Descrição
OnGetCellStyle	TsgcGridCellStyleEvent	Gerado por último para cada célula pintada, com o bloqueio dos dados mantido; Sender é o renderizador.
OnGetGroupText	TsgcGridGetGroupTextEvent	Altera o título de uma linha de grupo. Disparado com o bloqueio dos dados mantido.

Os procedimentos sgcGridRender, sgcGridSaveToPNG e sgcGridSaveToSVG fazem o mesmo com um renderizador temporário e um TsgcGridRenderOptions.
sgcGridColumnCaption retorna o título de uma coluna no painel de agrupamento e nas linhas de grupo: Caption, senão FieldName.

Exemplo

```
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;
```

Veja também

- Renderização headless
- Temas
- Formatação condicional
- Agrupamento
- Colunas Congeladas
- TsgcGridLineView

TsgcGridFormatRule

Uma regra de formatação condicional, um item de TsgcGridFormatRules, na unit sgcGrid_Format.

Propriedades

Propriedade	Tipo	Descrição
FieldName	string	A coluna cujo valor é testado.
Operator	TsgcGridFormatOperator	gfoEqual, gfoNotEqual, gfoGreater, gfoGreaterEqual, gfoLess, gfoLessEqual, gfoBetween, gfoContains, gfoStartsWith ou gfoIsEmpty.
Value, Value2	string	O valor com o qual comparar e o limite superior de gfoBetween (inclusive).
BackColor, ForeColor	TsgcUIColor	As cores de uma célula que atende à regra; 0 as deixa inalteradas.
Bold	Boolean	Texto em negrito para uma célula que atende à regra.
ApplyToRow	Boolean	Aplica o estilo a todas as células da linha quando o valor de FieldName atende à regra.
Enabled	Boolean	True por padrão.
StopOnMatch	Boolean	Uma correspondência encerra a avaliação dessa célula.

Métodos

Método	Retorna	Descrição
Matches(aValue, aDataType)	Boolean	Testa um valor.

Exemplo

```
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;
```

Veja também

- Formatação condicional
- TsgcGridFormatRules

TsgcGridFormatRules

A coleção de itens TsgcGridFormatRule, publicada como FormatRules pelas grades e por TsgcGridRenderOptions.

Métodos

Método	Retorna	Descrição
Add	TsgcGridFormatRule	Adiciona uma regra vazia.
Add(aFieldName, aOperator, aValue, aBackColor, aForeColor, aBold)	TsgcGridFormatRule	Adiciona uma regra em uma única chamada; aForeColor e aBold são opcionais.
Prepare, EvaluateRow, ApplyToCell		Usados pelo renderizador enquanto pinta.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Uma regra mudou.

Veja também

- Formatação condicional
- TsgcGridFormatRule

IsgcGridDataHost and IsgcUITable

IsgcGridDataHost

Uma interface da unit sgcGrid_Data com um único método, GetGridData, implementada por todo componente que possui um TsgcGridData: TsgcVCLGrid e TsgcFMXGrid. As fontes de dados e a consulta de IA encontram os dados de sua propriedade **Grid** com sgcGridResolveData: primeiro por meio de IsgcGridDataHost, que funciona em todas as versões do Delphi, inclusive o Delphi 7, depois por meio de uma propriedade publicada Data e, a partir do Delphi 2010, de uma propriedade pública Data. A propriedade GridData, quando atribuída, sempre prevalece sobre Grid.

IsgcUITable

A interface de tabela do sgcUICore (unit sgcUICore_Table), implementada por TsgcGridData e delegada a ele pelas grades. Ela permite que outro produto leia a grade sem depender do sgcGrid, como faz o TsgcChartDataSource_Table do sgcCharts. As linhas são linhas da exibição e as colunas são índices de Columns.

Método	Descrição
TableRowCount, TableColCount	O número de linhas da exibição e de colunas.
TableColName(aCol)	O FieldName de uma coluna.
TableValue(aRow, aCol), TableText(aRow, aCol)	Cells e CellText.
TableLock, TableUnlock	Lock e Unlock.
TableVersion	Incrementado a cada alteração.
TableAddListener, TableRemoveListener	Notificam os eventos gerados após OnChange, na thread da alteração, com os dados como Sender.

A contagem de referências está desabilitada. Os dados pertencem à grade, portanto libere a referência da interface antes de liberar a grade.

Exemplo

```
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;
```

Veja também

- Gráficos a partir da grade
- TsgcGridData

TsgcGridSource_DataSet

Preenche uma grade a partir de qualquer TDataSet, na unit sgcGrid_Source_DataSet (somente a unit DB, sem dependência de VCL ou FMX).

Propriedades

Propriedade	Tipo	Descrição
Grid	TComponent	A grade a preencher. Consulte IsgcGridDataHost.
GridData	TsgcGridData	Um data a preencher no lugar de uma grade. Pública; prevalece sobre Grid.
DataSet	TDataSet	O dataset de origem.
KeyField	string	O campo cujo texto é a chave da linha. Com uma chave, as linhas são atualizadas no local e mantêm seu flash, as chaves novas são acrescentadas e as ausentes são excluídas. Sem chave, cada carga substitui as linhas.
AutoColumns	Boolean	Cria as colunas a partir dos campos quando a grade não tem nenhuma. True por padrão.
LiveUpdates	Boolean	Acompanha o dataset: um Post, uma inserção, uma exclusão, um cancelamento ou uma atualização ressincroniza a grade. Edições não gravadas e a rolagem são ignoradas. True por padrão.
Active	Boolean	Carrega o dataset quando ele está aberto e continua a acompanhá-lo.

Métodos

Método	Retorna	Descrição
Load		Carrega o dataset agora.
Refresh		Recarrega quando Active.
FieldTypeToDataType(aFieldType)	TsgcGridDataType	O tipo de coluna de um tipo de campo. Função de classe.

Exemplo

```
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;
```

Veja também

- Vinculação de DataSet
- JSON e CSV

TsgcGridFeedSource

A classe base das fontes de feed do sgcGridWS, na unit sgcGridWS_Source. Ela lê cada mensagem como JSON e grava suas linhas em uma grade pela chave: somente os caminhos presentes em uma linha são gravados, de modo que uma mensagem pode trazer uma atualização parcial. Números JSON são gravados como Double, booleanos como Boolean, strings como string e null como Null.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Propriedades

Propriedade	Tipo	Descrição
Grid	TComponent	A grade na qual gravar.
GridData	TsgcGridData	Um data no qual gravar no lugar de uma grade. Pública; prevalece sobre Grid.
RowsPath	string	O caminho das linhas dentro de uma mensagem: um array de objetos de linha ou um único objeto de linha. Vazio: a própria mensagem é uma linha ou um array de linhas.
KeyPath	string	O caminho da chave da linha dentro de um objeto de linha, por exemplo symbol ou id.value. Uma linha sem chave é ignorada e informada por OnError.
Fields	TStrings	Um FieldName=json.path por linha, como Price=quote.last ou Bid=book.bid[0]. Uma linha sem = lê o caminho de mesmo nome. Vazio: cada coluna é lida do caminho com o mesmo nome do seu FieldName.
DeletePath	string	Um caminho booleano no objeto de linha: true, um número diferente de zero ou a string true exclui a linha da chave.
MaxRowsPerMessage	Integer	O máximo de linhas aplicadas a partir de uma mensagem (10000); 0 sem limite. As linhas excedentes são descartadas e informadas.
Active	Boolean	False ignora todas as mensagens. True por padrão.

Eventos

Evento	Tipo	Descrição
OnRow	TsgcGridFeedRowEvent	Aceita ou ignora a linha de uma chave.
OnError	TsgcGridFeedErrorEvent	Uma mensagem ou uma linha não pôde ser aplicada.

Threads

Todas as linhas de uma mensagem são aplicadas dentro de um único Data.BeginUpdate e EndUpdate, na thread em que a mensagem chega. OnRow e OnError também são executados nessa thread, portanto sincronize antes de mexer em um controle ali.

Exemplo

```
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;
```

Veja também

- TsgcGridFeedSource_WebSocket
- TsgcGridFeedSource_MQTT
- TsgcGridFeedSource_SSE
- Feeds em tempo real

TsgcGridFeedSource_WebSocket

Um TsgcGridFeedSource que lê as mensagens de um TsgcWebSocketClient do sgcWebSockets, na unit sgcGridWS_Source_WebSocket.
Enquanto a fonte está Active e Client está atribuído, em tempo de execução, ela instala o próprio manipulador OnMessage no cliente e preserva o manipulador que existia antes: cada mensagem de texto é aplicada primeiro à grade e depois repassada ao manipulador anterior, de modo que o manipulador da aplicação continua sendo disparado. Definir Active como False, limpar Client ou destruir a fonte restaura o manipulador anterior.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Propriedades

Propriedade	Tipo	Descrição
Client	TsgcWebSocketClient	O cliente WebSocket.
Grid	TComponent	A grade na qual gravar.
GridData	TsgcGridData	Um data no qual gravar no lugar de uma grade. Pública; prevalece sobre Grid.
RowsPath	string	O caminho das linhas dentro de uma mensagem: um array de objetos de linha ou um único objeto de linha. Vazio: a própria mensagem é uma linha ou um array de linhas.
KeyPath	string	O caminho da chave da linha dentro de um objeto de linha, por exemplo symbol ou id.value. Uma linha sem chave é ignorada e informada por OnError.
Fields	TStrings	Um FieldName=json.path por linha, como Price=quote.last ou Bid=book.bid[0]. Uma linha sem = lê o caminho de mesmo nome. Vazio: cada coluna é lida do caminho com o mesmo nome do seu FieldName.
DeletePath	string	Um caminho booleano no objeto de linha: true, um número diferente de zero ou a string true exclui a linha da chave.
MaxRowsPerMessage	Integer	O máximo de linhas aplicadas a partir de uma mensagem (10000); 0 sem limite. As linhas excedentes são descartadas e informadas.
Active	Boolean	False ignora todas as mensagens. True por padrão.

Métodos

Método	Retorna	Descrição
ProcessMessage(aText)	Integer	Aplica manualmente uma mensagem e retorna o número de linhas aplicadas.

Eventos

Evento	Tipo	Descrição
OnRow	TsgcGridFeedRowEvent	Aceita ou ignora a linha de uma chave.
OnError	TsgcGridFeedErrorEvent	Uma mensagem ou uma linha não pôde ser aplicada.

Exemplo

```
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;
```

Veja também

- TsgcGridFeedSource
- Feeds em tempo real

TsgcGridFeedSource_MQTT

Um TsgcGridFeedSource que lê as mensagens de um TsgcWSPClient_MQTT do sgcWebSockets, na unit sgcGridWS_Source_MQTT.
Enquanto a fonte está Active e Client está atribuído, em tempo de execução, ela instala o próprio manipulador OnMQTTPublish no cliente e preserva o manipulador que existia antes: cada publicação cujo tópico corresponde a Topic é aplicada primeiro à grade e depois repassada ao manipulador anterior, de modo que o manipulador da aplicação continua sendo disparado. Definir Active como False, limpar Client ou destruir a fonte restaura o manipulador anterior. A fonte não assina tópicos: a aplicação assina no cliente.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Propriedades

Propriedade	Tipo	Descrição
Client	TsgcWSPClient_MQTT	O cliente MQTT.
Topic	string	Um filtro de tópico MQTT: + corresponde a um nível, # a este nível e a todos os inferiores. Vazio aceita todas as publicações.
Grid	TComponent	A grade na qual gravar.
GridData	TsgcGridData	Um data no qual gravar no lugar de uma grade. Pública; prevalece sobre Grid.
RowsPath	string	O caminho das linhas dentro de uma mensagem: um array de objetos de linha ou um único objeto de linha. Vazio: a própria mensagem é uma linha ou um array de linhas.
KeyPath	string	O caminho da chave da linha dentro de um objeto de linha, por exemplo symbol ou id.value. Uma linha sem chave é ignorada e informada por OnError.
Fields	TStrings	Um FieldName=json.path por linha, como Price=quote.last ou Bid=book.bid[0]. Uma linha sem = lê o caminho de mesmo nome. Vazio: cada coluna é lida do caminho com o mesmo nome do seu FieldName.
DeletePath	string	Um caminho booleano no objeto de linha: true, um número diferente de zero ou a string true exclui a linha da chave.
MaxRowsPerMessage	Integer	O máximo de linhas aplicadas a partir de uma mensagem (10000); 0 sem limite. As linhas excedentes são descartadas e informadas.
Active	Boolean	False ignora todas as mensagens. True por padrão.

Métodos

Método	Retorna	Descrição
ProcessPublish(aTopic, aText)	Integer	Aplica uma publicação manualmente.

Eventos

Evento	Tipo	Descrição
OnRow	TsgcGridFeedRowEvent	Aceita ou ignora a linha de uma chave.
OnError	TsgcGridFeedErrorEvent	Uma mensagem ou uma linha não pôde ser aplicada.

Exemplo

```
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;
```

Veja também

- TsgcGridFeedSource
- Feeds em tempo real

TsgcGridFeedSource_SSE

Um TsgcGridFeedSource que lê as mensagens dos server-sent events de um TsgcHTTP1Client do sgcWebSockets, na unit sgcGridWS_Source_SSE.
Enquanto a fonte está Active e Client está atribuído, em tempo de execução, ela instala o próprio manipulador OnSSEMessage no cliente e preserva o manipulador que existia antes: cada trecho do fluxo é aplicado primeiro à grade e depois repassado ao manipulador anterior, de modo que o manipulador da aplicação continua sendo disparado. Definir Active como False, limpar Client ou destruir a fonte restaura o manipulador anterior. Os trechos são divididos em eventos: as linhas data são unidas, uma linha vazia encerra o evento e um evento pode se estender por vários trechos. A aplicação inicia o fluxo com Client.GetSSE.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Propriedades

Propriedade	Tipo	Descrição
Client	TsgcHTTP1Client	O cliente HTTP.
Grid	TComponent	A grade na qual gravar.
GridData	TsgcGridData	Um data no qual gravar no lugar de uma grade. Pública; prevalece sobre Grid.
RowsPath	string	O caminho das linhas dentro de uma mensagem: um array de objetos de linha ou um único objeto de linha. Vazio: a própria mensagem é uma linha ou um array de linhas.
KeyPath	string	O caminho da chave da linha dentro de um objeto de linha, por exemplo symbol ou id.value. Uma linha sem chave é ignorada e informada por OnError.
Fields	TStrings	Um FieldName=json.path por linha, como Price=quote.last ou Bid=book.bid[0]. Uma linha sem = lê o caminho de mesmo nome. Vazio: cada coluna é lida do caminho com o mesmo nome do seu FieldName.
DeletePath	string	Um caminho booleano no objeto de linha: true, um número diferente de zero ou a string true exclui a linha da chave.
MaxRowsPerMessage	Integer	O máximo de linhas aplicadas a partir de uma mensagem (10000); 0 sem limite. As linhas excedentes são descartadas e informadas.
Active	Boolean	False ignora todas as mensagens. True por padrão.

Métodos

Método	Retorna	Descrição
ProcessChunk(aText)	Integer	Aplica manualmente um trecho do fluxo.
ResetStream		Descarta um evento recebido parcialmente.

Eventos

Evento	Tipo	Descrição
OnRow	TsgcGridFeedRowEvent	Aceita ou ignora a linha de uma chave.
OnError	TsgcGridFeedErrorEvent	Uma mensagem ou uma linha não pôde ser aplicada.

Exemplo

```
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;
```

Veja também

- TsgcGridFeedSource
- Feeds em tempo real

TsgcGridAIQuery

Transforma uma pergunta em linguagem natural em filtros, ordenações e destaques de uma grade, na unit sgcGridWS_AI. A pergunta é resolvida pelo TsgcHTMLAIAssistant do sgcWebSockets: um LLM quando o Assistant tem um provedor, ou então o seu analisador baseado em regras, que entende perguntas como price greater than 100, sort by volume descending e highlight change less than 0. A resposta é conferida com as colunas da grade antes que qualquer coisa seja aplicada.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Propriedades

Propriedade	Tipo	Descrição
Grid	TComponent	A grade a consultar. Consulte IsgcGridDataHost.
GridData	TsgcGridData	Um data a consultar no lugar de uma grade. Pública; prevalece sobre Grid.
Assistant	TsgcHTMLAI-Assistant	O assistente que responde. Nil usa o analisador baseado em regras, sem rede e sem chave de API.
Active	Boolean	True por padrão.

Métodos

Método	Retor-na	Descrição
Ask(aQuestion)	Boole-an	Remove o que o Ask anterior aplicou e depois aplica a resposta. Síncrono; chame-o a partir da thread principal.
Reset		Restaura o OnFilterRow e a ordenação anteriores e exclui somente as regras de formatação que este componente adicionou.
LastExplanati-on	string	O que foi aplicado e o que foi ignorado.

Eventos

Evento	Tipo	Descrição
OnApplied	TNotifyEvent	Uma resposta foi aplicada.
OnError	TsgcGridAIErrorEvent	A pergunta não pôde ser respondida nem aplicada.

Exemplo

```
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;
```

Veja também

- Pergunte à sua grade
- Filtragem
- Formatação condicional

Atualizações em Tempo Real e Threads

O sgcGrid foi feito para linhas que mudam o tempo todo. Cada linha tem uma chave, e um escritor atualiza a linha de uma chave pelo nome do campo, a partir de qualquer thread, sem Synchronize.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com TsgcFMXGrid.

Atualização por chave

UpdateRow define os campos indicados da linha de uma chave e insere a linha quando a chave é nova (a menos que AutoInsert seja False). UpdateCell define um único campo, AppendRow adiciona uma linha com os valores na ordem das colunas e DeleteRow remove a linha de uma chave. Cada célula alterada registra o momento e a direção da alteração, para cima, para baixo ou alterada, o que aciona o flash ao alterar.

```
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;
```

Repinturas

Os dados geram OnChange na thread de escrita. A grade apenas se marca como suja ali e se repinta na thread principal no máximo MaxFPS vezes por segundo, portanto 1.000 atualizações por segundo custam no máximo 60 quadros. A linha selecionada é localizada novamente pela chave após cada alteração da exibição.

BeginUpdate e EndUpdate

Envolva um lote em BeginUpdate e EndUpdate para receber uma única notificação para todo ele.

```
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 de cauda

Para logs e feeds que crescem, **MaxRows** acima de 0 descarta as linhas mais antigas quando novas são inseridas, e **TailMode** mantém a linha mais nova visível enquanto a grade está rolada até o final. Rolar para cima interrompe o acompanhamento, e rolar de volta até o final o retoma.

```
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;
```

Veja também

- TsgcGridData
- Desempenho
- Feeds em tempo real

Ordenação

Cada coluna tem um SortOrder (gsoNone, gsoAscending ou gsoDescending) e um SortIndex, sua posição em uma ordenação por várias colunas. A ordenação é estável. O texto é comparado sem diferenciar maiúsculas de minúsculas e os demais tipos numericamente; valores nulos, vazios e não conversíveis sempre são ordenados primeiro.

Com o mouse

Um clique em um cabeçalho alterna a coluna entre crescente, decrescente e não ordenada. Com Ctrl pressionado, a coluna é adicionada à ordenação atual em vez de substituí-la, e o cabeçalho mostra a posição de cada coluna ordenada. OnSortChange é disparado após a alteração.

A partir do 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;
```

Regra de tempo real

Uma alteração de valor reordena a exibição somente quando sua coluna é uma coluna de ordenação ou quando há um filtro ativo. Uma coluna que muda o tempo todo e não está ordenada mantém a ordem atual e não custa nenhuma reordenação. Inserções, exclusões e alterações de ordenação ou de filtro sempre reconstroem a exibição.

Veja também

- Filtragem
- TsgcGridColumn

Filtragem

Dois filtros são aplicados, um após o outro. **FilterText** mantém as linhas cujo texto formatado, em qualquer coluna visível, contém o texto, sem diferenciar maiúsculas de minúsculas. **OnFilterRow** dos dados é então consultado para cada linha que passou.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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

O manipulador recebe uma linha do armazenamento, não uma linha da exibição: leia seus valores com StoreCells ou sua chave com StoreKeyOf. Ele é chamado com o bloqueio dos dados mantido, portanto não deve esperar por outra thread que use a grade. Quando a lógica do manipulador mudar, chame 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;
```

Com um filtro ativo, a alteração de qualquer valor pode mover uma linha para dentro ou para fora da exibição. RowOfKey retorna -1 para uma linha que foi filtrada.

Veja também

- Ordenação
- Pergunte à sua grade
- TsgcGridData

Agrupamento

O agrupamento organiza as linhas de uma grade sob linhas de grupo, um nível por coluna agrupada. Uma linha de grupo mostra o membro do seu nível, o número de linhas que contém e, com resumos, os totais delas. As linhas de grupo expandem e recolhem, e os níveis aninhados são recuados em GroupIndent DIPs (20 por padrão). O agrupamento faz parte de todas as edições, Community incluída, e funciona da mesma forma em TsgcVCLGrid e TsgcFMXGrid.
O agrupamento é estado da visão. A grade mantém suas linhas de grupo em um TsgcGridLineView (Lines), enquanto TsgcGridData, suas linhas da visão e IsgcUITable permanecem como estão, de modo que um gráfico ou qualquer outro leitor vinculado à grade continua vendo as linhas planas, ordenadas e filtradas. Sem coluna agrupada, a grade se comporta exatamente como uma grade sem agrupamento.

Agrupar uma coluna

Toda coluna tem um GroupIndex, -1 por padrão. Um valor de 0 ou mais agrupa a grade por essa coluna nesse nível: os níveis são ordenados por GroupIndex e depois pelo Index da coluna. Uma coluna oculta também agrupa. GroupBy define GroupIndex nas colunas indicadas (FieldName, sem diferenciar maiúsculas de minúsculas, nível 0 primeiro) e -1 em todas as outras; campos desconhecidos são ignorados. ClearGrouping define GroupIndex como -1 em todas as colunas. Alterar GroupIndex não reconstrói a visão dos dados.

- Os membros de um nível são ordenados pelo valor tipado da sua primeira linha, com a regra da ordenação da grade: texto sem diferenciar maiúsculas de minúsculas, os demais tipos numericamente, em ordem crescente, ou decrescente quando o SortOrder da coluna é gsoDescending.
- Null, texto vazio e valores que não são convertidos formam o membro vazio, mostrado como (blank) e sempre o primeiro.
- Quando a coluna tem um DisplayFormat, o membro é o texto formatado da célula, de modo que um DisplayFormat de yyyy-mm agrupa as datas por mês.
- Dentro de um grupo, as linhas mantêm a ordem da ordenação da grade e o filtro continua se aplicando.

O painel de agrupamento

ShowGroupPanel mostra um painel de GroupPanelHeight DIPs (36 por padrão) acima do cabeçalho, com um chip por nível agrupado, ou uma dica quando nada está agrupado. Com o mouse:

- Arraste o cabeçalho de uma coluna mais de 4 pixels até o painel para agrupar por ela no ponto onde soltar. Os cabeçalhos só são arrastados enquanto o painel está visível, e Escape cancela o arraste.
- Arraste um chip dentro do painel para mover seu nível, ou para fora do painel para desagrupar sua coluna.
- Clique em um chip para alternar a ordem de seus grupos. Isso inverte o SortOrder da coluna e dispara OnSortChange.

Em TsgcFMXGrid, um gesto de toque continua rolando a grade e cancela o arraste de um cabeçalho, de modo que o agrupamento por toque é feito por código ou pelo mouse.

Linhas de grupo

Uma linha de grupo mostra um chevron, o título e os resumos de grupo das colunas com um SummaryType (GroupSummaries, True por padrão). O título é o título da coluna, o membro e a contagem de linhas, como em Region: North (12). OnGetGroupText pode substituí-lo; ele é disparado com o bloqueio dos dados mantido, então leia a grade ali, mas nunca grave nos dados. GroupInfo(aGroup) retorna o nível, o grupo pai, a coluna agrupada, o texto e o valor do membro, a contagem de linhas, o estado de expansão e a linha de exibição de um grupo, veja TsgcGridLineView.

Expandir e recolher

GroupsExpanded (True por padrão) é o estado dos novos grupos; defini-lo mantém o estado dos grupos atuais. ExpandAll e CollapseAll alteram todos os grupos e os grupos criados depois. ExpandGroup e CollapseGroup alteram um grupo. OnGroupExpandChange informa o grupo e seu novo estado, com aGroup -1 depois de ExpandAll e CollapseAll. O estado de expansão de um grupo é mantido nas atualizações em tempo real e nas reordenações.
Um clique no chevron ou um clique duplo em uma linha de grupo alterna o grupo. Teclas em uma linha de grupo:

Tecla	Ação
Right	Expande o grupo, ou passa para a sua primeira linha filha quando ele já está expandido.
Left	Recolhe o grupo, ou passa para o seu grupo pai quando ele já está recolhido.
+ e -	Expandem e recolhem o grupo.
*	Expande o grupo e todos os grupos abaixo dele.
Enter	Alterna o grupo.
Ctrl+C	Copia o título da linha de grupo.

Quando um recolhimento oculta a linha selecionada, sua linha de grupo visível recebe o foco.

Linhas e seleção

Enquanto as linhas estão agrupadas, a grade rola linhas de exibição: TopRow e a barra de rolagem vertical contam linhas, e FocusedLine é a linha com o foco, -1 para nenhuma. Os índices de linha continuam sendo linhas da visão em todos os outros lugares (SelectedRow, Cells, OnCellClick). Em uma linha de grupo, SelectedRow é -1. A linha selecionada é mantida pela sua chave e encontrada de novo após um reagrupamento. GroupCount retorna o número de grupos e Lines fornece a visão completa das linhas.
Uma mudança de valor em tempo real em uma coluna agrupada move a linha para o novo grupo, e OnGroupingChange é disparado sempre que as colunas agrupadas ou seus níveis mudam, pelo painel ou por código.

Exemplo

Os manipuladores ficam em uma classe pequena, como os manipuladores dos outros exemplos de recursos.

```
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;
```

Veja também

- Resumos
- Colunas Congeladas
- Ordenação
- TsgcGridLineView
- TsgcGridColumn
- TsgcVCLGrid
- Agrupamento e Resumos do .NET

Resumos

Uma coluna com um SummaryType mostra um resumo de seus valores na linha de rodapé e em cada linha de grupo. Os resumos fazem parte de todas as edições, Community incluída.

SummaryType	Valor
gsuNone	Nenhum resumo (padrão).
gsuSum	A soma dos valores numéricos.
gsuCount	O número de valores que não são Null nem texto vazio.
gsuAverage	A soma dividida pelo número de valores numéricos.
gsuMin, gsuMax	O menor e o maior valor.

- Null, Unassigned e texto vazio são ignorados. O texto é convertido com o FormatSettings dos dados, como o modelo de dados o converte.
- As colunas de texto e booleanas suportam apenas gsuCount.
- As colunas de data mantêm Min e Max como datas e não têm Sum nem Average.
- Um resumo sem valor numérico é Null, e gsuCount é então 0.

Formato

SummaryFormat é um padrão de FormatFloat ou FormatDateTime. O texto literal vai entre aspas duplas, por exemplo `"Total: "#,##0.00`. Quando está vazio, usa-se o DisplayFormat da coluna, senão o padrão do tipo de dado; gsuCount usa `0`. Alterar SummaryType ou SummaryFormat atualiza os resumos sem reconstruir a visão dos dados.

Linhas de rodapé e de grupo

ShowFooter (False por padrão) mostra uma linha de rodapé de um RowHeight fixada na parte inferior da grade, com o resumo de cada coluna que tem um SummaryType sobre as linhas da visão, de modo que um filtro se aplica. O rodapé acompanha a rolagem horizontal e as colunas congeladas.
GroupSummaries (True por padrão) mostra os resumos de cada grupo, sobre as linhas do grupo, alinhados à direita em sua linha de grupo. A coluna visível mais à esquerda sempre contém o título: um resumo ali é desenhado depois do título quando cabe, senão só o rodapé o mostra. Veja Agrupamento.
FooterSummary(aCol) retorna o valor de rodapé de uma coluna (um índice de Columns), Null quando ela não tem nenhum, e FooterSummaryText(aCol) o seu texto formatado. Os valores dos grupos vêm de Lines.GroupSummary(aGroup, aCol). Os resumos são calculados uma vez e recalculados apenas quando os dados mudam, de modo que um quadro que os desenha não custa nada a mais. No modo virtual, os valores são lidos por OnGetValue para cada linha.

Exemplo

```
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;
```

Sem uma grade, TsgcGridSummaryAccumulator (unit sgcGrid_Summary) soma valores e retorna qualquer tipo de resumo com Value, TsgcGridSummaryCache guarda os valores de rodapé de um TsgcGridData, e sgcGridSummaryText formata um valor com as regras acima. O exemplo de TsgcGridLineView usa sgcGridSummaryText.

Veja também

- Agrupamento
- TsgcGridColumn
- TsgcGridLineView
- Agrupamento e Resumos do .NET

Linhas de Árvore

As linhas de árvore mostram as linhas de uma grade como uma hierarquia. Cada linha indica seu pai por meio de uma coluna de chave, e a grade dispõe as linhas em ordem de profundidade sob nós de árvore expansíveis, recuados em uma coluna. A árvore funciona com linhas mantidas na memória, linhas carregadas de um TDataSet e linhas alteradas por atualizações em tempo real, tanto em TsgcVCLGrid quanto em TsgcFMXGrid.
Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.

Ativar a árvore

A propriedade Tree da grade (TsgcGridTreeOptions, unit sgcGrid_Tree) guarda as configurações. O modo de árvore fica ativo quando KeyField e ParentField indicam ambos colunas da grade; limpe um deles para desativá-lo. As colunas de chave e de pai podem estar ocultas.

Propriedade	Padrão	Descrição
KeyField		A coluna que guarda a chave de cada linha.
ParentField		A coluna que guarda a chave da linha pai. Um valor vazio ou Null torna a linha uma raiz.
TreeField		A coluna que mostra o recuo e os botões da árvore. Vazio usa a primeira coluna visível.
Indent	16	O recuo de um nível, em DIPs.
Expanded	False	O estado dos nós que nunca foram expandidos nem recolhidos. Alterá-lo expande ou recolhe todos os nós.

Como as linhas são vinculadas

- As chaves são comparadas como texto, diferenciando maiúsculas de minúsculas. Null e texto vazio são a chave vazia. Inteiros e números de ponto flutuante sem fração usam seu texto inteiro, de modo que 5, 5.0 e '5' são a mesma chave. Os outros números de ponto flutuante e as datas usam o texto do FormatSettings dos dados.
- Os pais são procurados em todas as linhas armazenadas. Quando várias linhas compartilham uma chave, a primeira linha armazenada é dona dos filhos.
- Uma linha cujo pai é vazio, desconhecido (um órfão) ou a própria linha torna-se uma raiz. Um ciclo é quebrado na primeira linha que se repete, que se torna uma raiz.
- Os irmãos mantêm a ordem da ordenação da grade, de modo que ordenar uma coluna ordena os filhos de cada nó e as raízes.
- Com um filtro, uma linha visível cujo pai foi filtrado fica sob o seu ancestral visível mais próximo, ou torna-se uma raiz quando não há nenhum.
- Dados virtuais (Data.Virtual) não mostram árvore.

Expandir e recolher

ExpandNode(aRow) e CollapseNode(aRow) alteram o nó da linha da visão aRow; uma linha sem filhos fica como está. No modo de árvore, ExpandAll e CollapseAll atuam sobre os nós da árvore e sobre os nós criados depois. OnTreeExpandChange informa a linha da visão e seu novo estado, com aRow -1 depois de ExpandAll e CollapseAll. O estado é mantido por chave, de modo que sobrevive a uma ordenação, a um filtro, a um recarregamento das linhas e a atualizações em tempo real. Quando um recolhimento oculta a linha selecionada, seu pai visível mais próximo é selecionado.
Um clique em um botão da árvore alterna o nó. Teclas em uma linha selecionada no modo de árvore:

Tecla	Ação
Right	Expande o nó, ou passa para o seu primeiro filho quando ele já está expandido.
Left	Recolhe o nó, ou passa para o seu pai quando ele já está recolhido.
+ e -	Expandem e recolhem o nó.
*	Expande o nó e todos os nós abaixo dele.
Ctrl+Right e Ctrl+Left	Expandem e recolhem a faixa de detalhe da linha, veja Mestre Detalhe.

Em TsgcFMXGrid, um toque em um botão da árvore o alterna, e Cmd funciona como Ctrl.

Árvore e outros recursos

- O agrupamento é ignorado enquanto a árvore está ativa. O GroupIndex das colunas é mantido e volta a valer quando o modo de árvore é desativado.
- As faixas de detalhe funcionam no modo de árvore: cada linha mestre expandida mostra sua faixa abaixo dela.
- SaveToXLSX grava a árvore como uma estrutura de tópicos do Excel, veja Exportação para Excel.
- Os índices de linha continuam sendo linhas da visão (SelectedRow, Cells, OnCellClick). Lines.TreeMode informa se a árvore está ativa, e Lines.Tree (TsgcGridTreeBuilder) responde Depth, ParentRow, HasChildren, ChildCount e DescendantCount para uma linha da visão, veja TsgcGridLineView.

Exemplo

Um organograma mantido na memória. Os manipuladores ficam em uma classe pequena, como os manipuladores dos outros exemplos de recursos.

```
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;
```

Veja também

- Mestre Detalhe
- Agrupamento
- Exportação para Excel
- TsgcVCLGrid
- TsgcGridLineView
- Edições
- Linhas de Árvore e Mestre Detalhe do .NET

Mestre Detalhe

As linhas mestre-detalhe mostram uma faixa de detalhe sob cada linha mestre expandida. A faixa é uma pequena grade própria com as linhas relacionadas: os pedidos de um cliente, os itens de uma fatura. Ela tem cabeçalho, rola e ordena por conta própria. As faixas funcionam nos modos plano, agrupado e de árvore, tanto em TsgcVCLGrid quanto em TsgcFMXGrid.
Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.

Opções de detalhe

A propriedade Detail da grade (TsgcGridDetailOptions, unit sgcGrid_Detail) guarda as configurações.

Propriedade	Padrão	Descrição
Active	False	Ativa as faixas de detalhe.
Source	nil	O componente que guarda as linhas de detalhe: um TsgcGridDataStore ou uma grade. A grade o observa com uma free notification, de modo que uma origem liberada define Source como nil.
MasterKeyField		A coluna mestre cujo valor é procurado nas linhas de detalhe.
DetailKeyField		A coluna da origem que guarda a chave mestre.
Lines	8	A altura de uma faixa em linhas da grade, cabeçalho aninhado incluído. O menor valor é 2.
Columns		As colunas das grades de detalhe. Vazio usa uma cópia das colunas da origem. Armazenado apenas quando tem colunas.

Uma linha mostra um expansor de detalhe quando Active é True, MasterKeyField indica uma coluna mestre, e um Source ou um manipulador OnGetDetailData está definido. Uma linha cuja chave mestre é vazia nunca expande, e dados virtuais não mostram faixas.

De onde vêm as linhas

- OnGetDetailData é disparado primeiro, sem nenhum bloqueio mantido. Sender é a grade, aRow a linha mestre da visão, aMasterKey o texto de MasterKeyField e aData os dados da faixa. Preencha aData e defina aHandled como True para manter o que você gravou; o Source então não é lido. Sem Source, as colunas da faixa vêm de Detail.Columns.
- Caso contrário, a faixa é preenchida a partir do Source: as linhas armazenadas da origem cujo DetailKeyField corresponde à chave mestre, independentemente do filtro e da ordenação da origem. As linhas são copiadas por FieldName com suas chaves de origem. Uma coluna de detalhe sem campo na origem é Null, e uma origem virtual não fornece linhas.
- As chaves são comparadas como texto, diferenciando maiúsculas de minúsculas. Inteiros e números de ponto flutuante sem fração usam seu texto inteiro, de modo que 5, 5.0 e '5' são a mesma chave.
- Os dados da faixa adotam o FormatSettings da origem, ou o da grade mestre quando as linhas vêm apenas de OnGetDetailData.

Quando a origem muda, toda faixa aberta é preenchida de novo no próximo quadro. As linhas são atualizadas no lugar, de modo que uma célula alterada pisca como na grade mestre. OnGetDetailData é disparado de novo a cada preenchimento.

Expandir e recolher

ExpandDetail(aRow) e CollapseDetail(aRow) abrem e fecham a faixa da linha mestre da visão aRow, DetailExpanded(aRow) informa seu estado, e OnDetailExpandChange informa a linha e seu novo estado. DetailData(aRow) retorna os dados de uma faixa aberta, nil quando ela está recolhida. Eles são somente leitura; bloqueie-os enquanto os lê. O estado é mantido por chave mestre, de modo que sobrevive a uma ordenação, a um filtro, a um recarregamento das linhas mestre e a atualizações em tempo real.
Um clique no expansor alterna a faixa. Down pula as linhas de uma faixa. Teclas em uma linha selecionada:

Modo	Teclas
Plano ou agrupado, seleção de linha	Right e Left expandem e recolhem a faixa.
Plano ou agrupado, SelectionMode gsmCell	Ctrl+Right e Ctrl+Left expandem e recolhem a faixa; Right e Left continuam movendo entre as células.
Modo de árvore	Ctrl+Right e Ctrl+Left expandem e recolhem a faixa; Right e Left atuam sobre o nó da árvore.

Dentro de uma faixa

- A roda do mouse sobre uma faixa rola a faixa. No fim da faixa, a grade rola em seu lugar.
- Um clique no cabeçalho aninhado ordena as linhas da faixa. OnSortChange não é disparado.
- Um clique em uma célula aninhada a seleciona e dispara OnDetailCellClick com a linha mestre e a linha da visão e o índice de Columns dos dados de detalhe. A seleção mestre não se move.
- As grades de detalhe são somente leitura e têm um nível. As faixas não são gravadas por SaveToXLSX.

Em TsgcFMXGrid, a roda rola uma faixa apenas enquanto o ponteiro do mouse está sobre ela, um gesto de toque continua rolando a grade, um toque no expansor o alterna, e Cmd funciona como Ctrl.

Exemplo

Clientes com uma faixa de seus pedidos, mantidos em um TsgcGridDataStore preenchido 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;
```

As mesmas faixas com as linhas gravadas por OnGetDetailData, por exemplo a partir de uma 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 os pedidos de um TDataSet, carregue-o em um TsgcGridDataStore com um TsgcGridSource_DataSet, veja TsgcGridDataStore.

Veja também

- TsgcGridDataStore
- Linhas de Árvore
- Vinculação de DataSet
- TsgcVCLGrid
- Edições
- Linhas de Árvore e Mestre Detalhe do .NET

Seleção e Teclado

SelectionMode gsmRow seleciona linhas inteiras e gsmCell células individuais. SelectedRow e SelectedCol são a linha da exibição e o índice da coluna, e SelectedKey é a chave da linha selecionada: quando uma atualização em tempo real move a linha, a seleção a acompanha.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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	Ação
Seta para cima, seta para baixo	Linha anterior ou seguinte.
Page Up, Page Down	Uma página para cima ou para baixo.
Home, End	Primeira ou última linha; no modo de célula, a primeira ou a última coluna, e Ctrl+Home ou Ctrl+End a primeira ou a última linha.
Seta para a esquerda, seta para a direita	Coluna anterior ou seguinte no modo de célula.
Ctrl+C, Ctrl+Insert	Copia a linha selecionada como texto separado por tabulações, ou a célula selecionada no modo de célula. Cmd+C no macOS.

F2, Enter, um caractere, Espaço	Iniciam a edição quando a grade não é ReadOnly. Consulte Edição no local.

Mouse

Um clique seleciona, OnCellClick e OnCellDblClick informam a célula, a roda rola, e RowAt e ColAt localizam a célula sob um ponto.

Veja também

- TsgcVCLGrid
- Redimensionamento de colunas

Redimensionamento de Colunas

Arraste a borda direita de uma célula de cabeçalho para redimensionar a coluna; o cursor muda sobre a borda. A largura nunca fica abaixo de MinWidth da coluna, 20 por padrão. A nova largura é armazenada em Width da coluna e OnColumnResized é disparado.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Veja também

- TsgcGridColumn
- Seleção e teclado

Colunas Congeladas

FrozenColumns (0 por padrão) mantém as primeiras colunas visíveis no lugar enquanto as outras colunas rolam na horizontal, de modo que uma coluna de chave, como um símbolo ou um nome, permanece à vista. O cabeçalho, o rodapé e os resumos de grupo seguem a mesma divisão, e uma linha de borda de um pixel marca a borda congelada. As colunas congeladas fazem parte de todas as edições, Community incluída.

- FrozenColumns conta colunas visíveis: uma coluna oculta é ignorada.
- Os chevrons de grupo à esquerda (a margem, um GroupIndent por nível agrupado) também nunca rolam.
- Em TsgcVCLGrid, a barra de rolagem horizontal nativa abrange apenas a parte que rola, de modo que o maior ScrollX não muda. TsgcFMXGrid desenha seu indicador horizontal sobre a parte que rola da mesma forma.
- Os movimentos do teclado trazem para a vista uma coluna rolada à direita da borda congelada.

O renderizador expõe a mesma configuração em suas Options, de modo que a saída headless em PNG e SVG coincide com o controle, e FrozenWidth retorna a largura que não rola.

Exemplo

```
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;
```

Veja também

- Redimensionamento de Colunas
- Agrupamento
- Renderização Headless
- TsgcGridRenderer

Flash ao Alterar

Com **FlashChanges** True (o padrão), uma célula cujo valor mudou recebe um tom com a cor da sua direção: FlashUpColor quando o valor subiu, FlashDownColor quando desceu e FlashChangedColor para qualquer outra alteração. O tom desaparece linearmente ao longo de **FlashDuration** milissegundos, 800 por padrão.
FlashMode fmBackground tinge o fundo da célula e fmText o texto. Uma cor 0 usa o padrão do tema: verde, vermelho e o destaque, translúcidos para o fundo e cores de texto legíveis para fmText, mais claras em um tema escuro.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Enquanto um flash desaparece, a grade continua solicitando quadros, até MaxFPS. A seleção tem prioridade sobre o flash, e o flash sobre as regras de formatação.

Veja também

- Formatação condicional
- Temas

Formatação Condicional

As **FormatRules** de uma grade colorem as células cujo valor atende a uma condição. Uma regra compara o valor da coluna FieldName com Value, e com Value2 em gfoBetween, segundo o DataType dessa coluna: números numericamente, datas como datas (Value em ISO 8601 ou um número de série de data), booleanos como true ou false, texto sem diferenciar maiúsculas de minúsculas. gfoContains e gfoStartsWith sempre comparam texto, gfoIsEmpty corresponde a um valor nulo ou em branco, e uma célula vazia não corresponde a nenhum operador de ordenação.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Precedência

- As regras são avaliadas na ordem da coleção. Toda regra habilitada que corresponde aplica seus BackColor e ForeColor diferentes de zero, com uma correspondência posterior sobrepondo uma anterior, e Bold se acumula.
- Uma regra correspondente com StopOnMatch encerra a avaliação dessa célula.
- Uma regra ApplyToRow testa a própria coluna e, havendo correspondência, aplica o estilo a todas as células da linha.
- Quando a grade pinta, a seleção prevalece sobre o flash, o flash sobre as regras e as regras sobre o tom da linha alternada.
- OnGetCellStyle é gerado por último para cada célula pintada: aFore e aBold trazem o resultado até então, e aBack é um preenchimento extra sobre ele.

```
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;
```

Veja também

- TsgcGridFormatRule
- Flash ao alterar
- Pergunte à sua grade

Edição no Local

A edição no local precisa de SGC_GRID_EDIT, que a edição Community não define. Consulte Editions.

Iniciando uma edição

Defina **ReadOnly** como False. Uma coluna é editável enquanto seu Editable for True. F2, Enter, um caractere digitado (que substitui o texto) ou um clique duplo abre um TEdit sobre a célula, ou um TComboBox quando a coluna tem uma PickList. StartEdit o abre a partir do código, e OnBeforeEdit pode recusá-lo.

Confirmar e cancelar

Enter confirma, Tab e Shift+Tab confirmam e passam para a próxima ou a anterior célula editável, e Escape cancela. Perder o foco confirma, ou cancela quando o texto é inválido. CommitEdit e CancelEdit fazem o mesmo a partir do código.

Conversão e validação

O texto é convertido para o DataType da coluna com Data.FormatSettings. Um texto que não converte mantém o editor aberto e dispara OnEditError, cujo aMessage pode ser alterado. OnValidate recebe o valor convertido e pode alterá-lo ou rejeitá-lo. O valor é então gravado com Data.UpdateCell, de modo que o rastreamento de alterações, o flash e a ordenação se aplicam, e OnEdited informa o valor antigo e o novo.

Booleanos

Uma coluna gdtBoolean não tem editor: Espaço, um clique, F2, Enter ou StartEdit alternam o valor.

Atualizações em tempo real durante a edição

Quando uma atualização em tempo real reordena ou filtra a exibição, o editor acompanha a sua linha, localizada novamente pela chave. Quando a linha é excluída ou filtrada, a edição é cancelada.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Veja também

- TsgcGridColumn
- TsgcVCLGrid

Modo Virtual

No modo virtual, a grade não armazena linhas. **VirtualRowCount** define quantas linhas existem, e **OnGetValue** dos dados retorna o valor de uma linha e coluna quando o renderizador as pinta. Como somente as linhas visíveis são pintadas, uma grade de um milhão de linhas custa o mesmo que uma de cinquenta.
As linhas virtuais não têm chaves, portanto a exibição é as linhas em ordem: não há ordenação, filtragem nem rastreamento de alterações, e KeyOf e RowOfKey não retornam nada. Altere VirtualRowCount quando o número de linhas mudar.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Veja também

- TsgcGridData
- Desempenho

JSON e CSV

A unit sgcGrid_IO carrega e grava um TsgcGridData como JSON ou CSV, de e para strings, streams e arquivos. Ela é neutra em relação ao framework e compila também no Linux64.

Carregamento

- Uma carga substitui as linhas e é executada dentro de um único BeginUpdate e EndUpdate. Com um campo de chave, a chave de uma linha é o texto desse campo; uma chave repetida atualiza a linha anterior.
- Com aAutoColumns True e ainda sem colunas, as colunas são criadas a partir da entrada e seu DataType é deduzido dos valores. As colunas existentes são mantidas e associadas por FieldName.
- A entrada JSON é um array de objetos. Um FieldName com pontos, como price.last, lê membros aninhados.
- O CSV segue a RFC 4180, e o primeiro registro contém os FieldNames. aInvariant True (o padrão) lê e grava números com ponto e datas em ISO 8601; False usa o FormatSettings dos dados.

Gravação

GridSaveToJSON e GridSaveToCSV gravam as linhas na ordem da exibição, ordenadas e filtradas, com as colunas em ordem e somente as visíveis quando aVisibleOnly é True. O texto é UTF-8 sem BOM.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Veja também

- Vinculação de DataSet
- TsgcGridData

Exportação para Excel

SaveToXLSX grava a grade como uma pasta de trabalho do Excel (Office Open XML, uma planilha) em um stream ou em um arquivo, sem biblioteca externa. A pasta de trabalho mantém o que a grade mostra: as colunas visíveis na ordem de exibição com suas larguras e formatos de número atuais, o cabeçalho congelado e FrozenColumns, os grupos ou a árvore como uma estrutura de tópicos recolhível, e os resumos como fórmulas.
Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.

Exportar uma grade

Método	Descrição
SaveToXLSX(aStream; aSheetName = 'Sheet1')	Grava a pasta de trabalho em um stream.
SaveToXLSX(aFileName; aSheetName = 'Sheet1')	Grava a pasta de trabalho em um arquivo.

Chame-o na thread da UI. O nome da planilha perde os caracteres que o Excel rejeita (eles viram _) e é cortado em 31 caracteres. Uma planilha que precisaria de mais de 1,048,576 linhas ou 16,384 colunas dispara EsgcGridXLSXError antes de qualquer coisa ser gravada.

O que a planilha contém

- A linha 1 é o cabeçalho em negrito, congelado junto com as primeiras FrozenColumns colunas. As larguras das colunas são as larguras na tela.
- Cada linha de grupo e cada linha da visão é gravada, independentemente do que esteja recolhido. As linhas dentro de um grupo recolhido ficam ocultas e a linha de grupo é marcada como recolhida, de modo que o Excel abre a estrutura de tópicos como a grade a mostra. As linhas de grupo ficam acima de suas linhas.
- Uma linha de grupo guarda seu título (Region: North (12)) na primeira coluna, ou na primeira coluna sem resumo. OnGetGroupText não é usado.
- As linhas de árvore recebem o nível de estrutura de tópicos de sua profundidade e um recuo na coluna da árvore. Os níveis da estrutura de tópicos param em 7, o máximo do Excel.
- As faixas de detalhe não são exportadas.
- Os números são gravados como números, as datas como datas seriais do Excel, os booleanos como booleanos e o texto como strings compartilhadas sem caracteres de controle.
- O DisplayFormat de uma coluna torna-se o formato de número do Excel. Os padrões de FormatFloat passam como estão; nos padrões de data, nn vira mm, z vira 000 e am/pm vira AM/PM. Sem DisplayFormat, uma coluna de inteiros usa 0, uma coluna de moeda #,##0.00 e uma coluna de data yyyy-mm-dd hh:mm:ss.

Resumos

Cada coluna de resumo de uma linha de grupo guarda uma fórmula SUBTOTAL sobre as linhas do grupo, com o valor da grade em cache na célula, de modo que o arquivo mostra os totais antes de o Excel recalcular. Quando uma coluna visível tem um resumo, uma linha de rodapé guarda SUBTOTAL sobre todas as linhas de dados. A linha de rodapé é gravada qualquer que seja o valor de ShowFooter.

SummaryType	SUBTOTAL	Tipos de coluna
gsuSum	9	Números
gsuAverage	1	Números
gsuMin	5	Números e datas
gsuMax	4	Números e datas

gsuCount	3	Todos os tipos (células não vazias)

Esses códigos contam as linhas ocultas, de modo que um grupo recolhido mantém seu total, e SUBTOTAL ignora as células SUBTOTAL aninhadas dos grupos internos.

TsgcGridXLSXWriter

O gravador por trás de SaveToXLSX fica na unit sgcGrid_XLSX e não precisa de uma grade: ele grava qualquer TsgcGridData.

Membro	Descrição
SaveToStream(aData, aLines, aStream), SaveToFile(aData, aLines, aFileName)	Grava a pasta de trabalho de aData. aLines fornece o agrupamento, a árvore e o estado de expansão; nil usa uma visão de linhas própria.
SheetName	O nome da planilha, 'Sheet1' por padrão.
FrozenColumns	As primeiras colunas visíveis mantidas na vista.
SetColumnWidth(aCol, aDIPs)	A largura da coluna aCol (índice de Columns) em DIPs, usada no lugar de seu Width. Um valor negativo restaura Width.
Summaries	Grava as células SUBTOTAL de grupo e de rodapé, True por padrão.
FileTime	A data e a hora das entradas do zip, Now quando o gravador é criado. Os mesmos dados e o mesmo FileTime sempre dão os mesmos bytes.
sgcGridExcelNumberFormat(aDataType, aDisplayFormat)	O código de formato de número do Excel de um tipo de coluna e DisplayFormat.

Exemplo

```
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;
```

Veja também

- JSON e CSV
- Resumos
- Agrupamento
- Linhas de Árvore
- Colunas Congeladas
- Edições
- Exportação para Excel do .NET

Vinculação de DataSet

Um TsgcGridSource_DataSet preenche uma grade a partir de qualquer TDataSet. Com um KeyField, a grade é sincronizada em vez de recarregada: as linhas alteradas são atualizadas no local e piscam, as chaves novas são acrescentadas e as ausentes são excluídas. Com LiveUpdates, a grade acompanha cada alteração gravada no dataset.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

O dataset é percorrido com DisableControls e seu registro atual é restaurado. Fechar o dataset mantém as linhas. O demo DataSetGrid mostra a vinculação com um TClientDataSet.

Veja também

- TsgcGridSource_DataSet
- JSON e CSV
- Demos

Feeds em Tempo Real

As fontes de feed do sgcGridWS gravam mensagens JSON de clientes do sgcWebSockets diretamente em uma grade: TsgcGridFeedSource_WebSocket, TsgcGridFeedSource_MQTT e TsgcGridFeedSource_SSE.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Regras de mapeamento

Propriedade	Regra
RowsPath	Onde estão as linhas em uma mensagem. Vazio: a mensagem é um objeto de linha ou um array de linhas.
KeyPath	O caminho da chave dentro de uma linha, como symbol ou id.value. Uma string, um número ou um booleano é aceito.
Fields	Linhas FieldName=json.path; uma linha sem = lê o caminho de mesmo nome; vazio lê cada coluna do caminho com o mesmo nome do seu FieldName. Somente os caminhos presentes são gravados, portanto atualizações parciais funcionam.
DeletePath	Um caminho booleano que exclui a linha da chave quando é 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;
```

Server-sent events

```
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;
```

Mensagens de outras origens

```
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;
```

As linhas de uma mensagem são aplicadas em um único BeginUpdate e EndUpdate na thread do cliente, o que é seguro porque os dados são seguros para threads. O demo LiveFeed mostra a fonte WebSocket com um servidor no mesmo processo.

Veja também

- TsgcGridFeedSource
- Atualizações em tempo real e threads

Pergunte à Sua Grade

TsgcGridAIQuery permite que o usuário filtre, ordene e destaque uma grade com uma pergunta em linguagem natural, como price greater than 100 sort by volume descending highlight change less than 0.
Este recurso faz parte da ponte sgcGridWS: as units precisam dos fontes ou dos pacotes do sgcWebSockets, e a edição Community não o inclui.

Como a resposta é aplicada

- Os filtros são combinados com AND e aplicados por meio de OnFilterRow dos dados. Um manipulador já atribuído é preservado e chamado primeiro.
- As ordenações substituem SortOrder e SortIndex das colunas.
- Os destaques são adicionados às FormatRules da grade com ApplyToRow definido.
- Agrupamento e limites de linhas não são suportados pela grade; eles são ignorados e informados em LastExplanation.
- Cada Ask chama Reset primeiro, portanto as perguntas não se acumulam. Reset restaura o manipulador de filtro e a ordenação anteriores e remove somente as regras que a consulta adicionou.

```
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;
```

Com um LLM

Atribua um TsgcHTMLAIAssistant com um provedor para responder com um LLM. Ask é síncrono e bloqueia enquanto o provedor responde, portanto chame-o a partir da thread principal. Sem um Assistant, o analisador baseado em regras responde, sem rede e sem chave de API, como no 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;
```

Veja também

- TsgcGridAIQuery
- Filtragem
- Formatação condicional

Gráficos a Partir da Grade

Os dados de uma grade implementam IsgcUITable, a interface de tabela do sgcUICore, e as grades a delegam. A fonte de dados **TsgcChartDataSource_Table** do sgcCharts lê essa tabela: defina seu Table como a grade e XColumn e YColumn como FieldNames de colunas. Toda alteração nas linhas da grade, uma atualização vinda de um feed ou uma edição no local, chega ao gráfico sem nenhum código intermediário, e o gráfico acompanha a ordenação e o filtro da grade.

```
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;
```

O exemplo precisa das units do sgcCharts. O demo GridAndChart coloca a grade e o gráfico em dois blocos do sgcDashboard.

Veja também

- IsgcGridDataHost e IsgcUITable
- Demos

Temas

A grade usa os tokens de tema claro e escuro do sgcUICore: Theme utLight (padrão) ou utDark, AccentColor para a seleção e o flash de alteração, e FontSize, RowHeight, HeaderHeight, AlternateRows e as linhas de grade para a densidade. As cores são valores ARGB TsgcUIColor; sgcUIARGB cria uma.
Um TsgcVCLGrid colocado em um bloco do sgcDashboard acompanha o tema do dashboard por meio de sua propriedade publicada Theme.
Os exemplos são procedimentos simples que recebem um **TsgcVCLGrid**. Os manipuladores de eventos são métodos de uma pequena classe, TGridHandlers, declarada em TsgcVCLGrid. O mesmo código funciona com 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;
```

Veja também

- Flash ao alterar
- Renderização headless

Renderização Headless

O renderizador não precisa de uma janela. SaveToPNG e SaveToSVG das grades gravam a exibição atual; os procedimentos sgcGridSaveToPNG e sgcGridSaveToSVG, ou um TsgcGridRenderer, renderizam um TsgcGridData independente, inclusive no Linux64, para relatórios, e-mails ou imagens geradas no servidor. A imagem da Introdução foi criada pelo exemplo abaixo.

```
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;
```

A partir de uma grade

```
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;
```

Veja também

- TsgcGridRenderer
- Temas

Visão Geral do .NET

sgcGrid .NET é uma porta gerenciada em C# da biblioteca sgcGrid do Delphi, e não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcGrid** e mantêm os nomes do Delphi, portanto **TsgcGridData**, **TsgcGridColumns**, **TsgcGridColumn**, **TsgcGridFormatRules**, **TsgcGridRenderer** e **TsgcGridSource_DataSet** se leem da mesma forma nas duas linguagens. Os controles de grade seguem o padrão de nomes do Delphi: **TsgcWinFormsGrid** para WinForms, **TsgcWpfGrid** para WPF e **TsgcAvaloniaGrid** para Avalonia. Os três desenham com a porta .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe, de modo que a grade tem a mesma aparência em todos os hosts e na saída headless em PNG e SVG.
A grade foi criada para dados ao vivo. O armazenamento de linhas é indexado por chave e seguro para threads, então uma thread de trabalho pode atualizar, acrescentar e excluir linhas enquanto a grade desenha apenas as linhas visíveis, em uma taxa de quadros controlada. As células alteradas piscam, as regras de formatação colorem células ou linhas inteiras, e a grade ordena, filtra, funciona em modo virtual e exporta para JSON e CSV. Na edição Registered o usuário edita as células no local, a ponte com o sgcWebSockets alimenta a grade a partir de conexões WebSocket, MQTT e SSE, e a ponte com o sgcHTML responde a perguntas em linguagem natural sobre as linhas.

Pacotes

A porta é dividida em pacotes NuGet, de modo que um projeto só inclui o que usa.

PackageId	Conteúdo	Frameworks de destino
esegece.sgcGrid	O modelo (**TsgcGridColumns**, **TsgcGridData**, **TsgcGridFormatSettings**), ordenação e filtragem, modo virtual, flash e regras de formatação (**TsgcGridFormatRules**), o renderizador **TsgcGridRenderer** com a saída SVG headless, importação e exportação de JSON e CSV (**sgcGrid_IO**) e a fonte DataTable **TsgcGridSource_DataSet**. Sem dependência nativa.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGrid.Rendering	A saída PNG headless (**SaveToPNG**), por meio do 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**, no Avalonia 11.3.	net8.0, net9.0
esegece.sgcGrid.sgcWebSockets	A ponte de feeds em tempo real: **TsgcGridFeedSource**, **TsgcGridFeedSource_WebSocket**, **TsgcGridFeedSource_MQTT** e	netstandard2.0,

	TsgcGridFeedSource_SSE. Depende do pacote esegece.sgcWebSockets. Somente na edição Registered.	net8.0, net9.0
esegece.sgcGrid.sgcHTML	A ponte de consultas por IA: **TsgcGridAIQuery** e **TsgcGridAIFilterItem**, sobre o **TsgcHTMLAIAssistant** do pacote esegece.sgcHTML. Somente na edição Registered.	netstandard2.0, net8.0, net9.0

Cada pacote de host depende do esegece.sgcGrid.Rendering e do pacote sgcUICore do mesmo host, portanto basta adicionar o pacote do host.

Edições

Edição	Ids dos pacotes	Inclui
Registered	esegece.sgcGrid, .Rendering, .WinForms, .Wpf, .Avalonia, .sgcWebSockets e .sgcHTML	Tudo: as três grades, atualizações em tempo real, flash e regras de formatação, ordenação, filtragem, modo virtual, JSON e CSV, a fonte DataTable, a saída headless, a edição no local e as duas pontes.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcGrid.WinForms.Community. Não há pacotes de ponte Community.	Tudo, exceto a edição no local (**StartEdit**, **CommitEdit**, **CancelEdit**, os eventos **OnBeforeEdit**, **OnValidate**, **OnEdited** e **OnEditError**) e as duas pontes.

É a mesma divisão das edições do Delphi, veja Edições.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	TsgcVCLGrid	Windows, no .NET Framework 4.7.2 e no .NET 8 e 9.
WPF	TsgcFMXGrid	Windows, no .NET Framework 4.7.2 e no .NET 8 e 9.
Avalonia	TsgcFMXGrid	Windows, Linux e macOS, no .NET 8 e 9.
Headless	TsgcGridRenderer	Qualquer plataforma do .NET Standard 2.0, inclusive servidores Linux.

Veja também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em Relação ao Delphi
- Renderização Headless no .NET
- Vinculação de DataTable, JSON e CSV no .NET
- Edição no Local no .NET
- Feeds em Tempo Real no .NET
- Pergunte à Sua Grade no .NET
- Desempenho no .NET
- Demos do .NET

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, e os pacotes core e de renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcGrid.WinForms
dotnet add package esegece.sgcGrid.Wpf
dotnet add package esegece.sgcGrid.Avalonia
```

Adicione as pontes para feeds em tempo real e consultas em linguagem natural. Elas trazem consigo os pacotes esegece.sgcWebSockets e esegece.sgcHTML:

```
dotnet add package esegece.sgcGrid.sgcWebSockets
dotnet add package esegece.sgcGrid.sgcHTML
```

Edição Community

```
dotnet add package esegece.sgcGrid.WinForms.Community
dotnet add package esegece.sgcGrid.Wpf.Community
dotnet add package esegece.sgcGrid.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcGrid.WinForms" Version="2026.10.0" />
</ItemGroup>
```

Em um projeto, referencie os pacotes Registered ou os Community, nunca os dois: eles contêm os mesmos tipos no mesmo namespace. O código que usa a API de edição não compila com os pacotes Community; proteja-o com `#if !SGC_COMMUNITY` quando um mesmo projeto compilar as duas edições.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:grid="clr-namespace:esegece.sgcGrid;assembly=esegece.sgcGrid.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como em qualquer aplicativo desktop Avalonia. Mapeie o namespace no XAML com `xmlns:grid="using:esegece.sgcGrid"`.
- Headless: o esegece.sgcGrid sozinho basta para o modelo, a fonte DataTable, JSON, CSV e SVG. Adicione o esegece.sgcGrid.Rendering para PNG, que traz o SkiaSharp e seus recursos nativos para Windows e Linux.
- Pontes: o esegece.sgcGrid.sgcWebSockets e o esegece.sgcGrid.sgcHTML têm como destino netstandard2.0, net8.0 e net9.0, portanto funcionam com todos os hosts, inclusive o .NET Framework 4.7.2.

Em C#, todo host precisa de `using esegece.sgcGrid;`, e de `using esegece.sgcUICore;` para o **TsgcUITheme** e os demais tipos dos temas. As fontes de feed em tempo real também precisam de `using esegece.sgcWebSockets;` para os clientes de socket.

Veja também

- Visão Geral do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia

- Feeds em Tempo Real no .NET
- Pergunte à Sua Grade no .NET
- Demos do .NET

Início Rápido WinForms

Este exemplo cria um formulário WinForms com uma grade de cotações. Uma thread de trabalho atualiza as linhas quatro vezes por segundo, as células alteradas piscam, uma regra de formatação pinta em vermelho claro as linhas com variação negativa, e uma caixa de seleção alterna a grade entre o tema claro e o escuro.

Passos

- Crie um projeto WinForms `net8.0-windows` e adicione o pacote **esegece.sgcGrid.WinForms**, veja Instalação pelo NuGet.
- Crie um **TsgcWinFormsGrid** e ancore-o. Adicione as colunas com **Columns.Add**: o nome do campo, o título e o **TsgcGridDataType**, e defina **DisplayFormat** para números e datas.
- Adicione as linhas com **Data.AppendRow**: o primeiro argumento é a chave da linha e os valores seguem a ordem das colunas.
- Atualize as linhas pela chave com **Data.UpdateRow** ou **Data.UpdateCell**, a partir de qualquer thread. **BeginUpdate** e **EndUpdate** agrupam várias alterações em um único redesenho.
- Adicione uma regra a **FormatRules** para colorir uma célula, ou a linha inteira com **ApplyToRow**.

Exemplo

As cores são valores `$AARRGGBB`; um alfa abaixo de `FF` tinge a célula sobre o seu fundo. A grade é redesenhada no máximo **MaxFPS** vezes por segundo, por mais rápido que os dados mudem, veja Atualizações em Tempo Real e Threads e Desempenho no .NET.

Veja também

- Instalação pelo NuGet
- Início Rápido WPF
- Início Rápido Avalonia
- Classes do .NET
- TsgcVCLGrid
- Flash ao Alterar
- Formatação Condicional
- Demos do .NET

Início Rápido WPF

A mesma grade de cotações em WPF. A grade e suas opções de exibição são declaradas em XAML; as colunas e as linhas são definidas no code-behind.

Passos

- Crie um projeto WPF `net8.0-windows` e adicione o pacote **esegece.sgcGrid.Wpf**, veja Instalação pelo NuGet.
- Mapeie o namespace na janela com `xmlns:grid="clr-namespace:esegece.sgcGrid;assembly=esegece.sgcGrid.Wpf"`.
- Declare um **TsgcWpfGrid** e defina **RowHeight**, **FlashChanges**, **TailMode** e as demais opções de exibição como atributos.
- Adicione as colunas, as regras de formatação e as linhas no construtor da janela, depois de `InitializeComponent`.

Exemplo

MainWindow.xaml:
MainWindow.xaml.cs:
O App.xaml é o de um novo projeto WPF, com `StartupUri="MainWindow.xaml"`. A grade WPF desenha suas próprias barras de rolagem, como faz a grade FireMonkey, veja Diferenças do .NET em Relação ao Delphi.

Veja também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em Relação ao Delphi
- TsgcFMXGrid
- Demos do .NET

Início Rápido Avalonia

A mesma grade de cotações em Avalonia, no Windows, Linux e macOS.

Passos

- Crie um projeto desktop Avalonia `net8.0` com **Avalonia.Desktop** e **Avalonia.Themes.Fluent** 11.3, e adicione o pacote **esegece.sgcGrid.Avalonia**, veja Instalação pelo NuGet.
- Mapeie o namespace na janela com `xmlns:grid="using:esegece.sgcGrid"`.
- Declare um **TsgcAvaloniaGrid** com um `Name`, localize-o depois de `AvaloniaXamlLoader.Load` e adicione as colunas e as linhas.
- Alterne o **Theme** a partir do code-behind.

Exemplo

MainWindow.axaml:
MainWindow.axaml.cs:
Program.cs:
Em uma tela sensível ao toque, o conteúdo acompanha o dedo e um toque seleciona a célula. Por meio de uma referência do tipo `Control`, **Theme** é o tema de controle do Avalonia, veja Diferenças do .NET em Relação ao Delphi.

Veja também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Classes do .NET
- Diferenças do .NET em Relação ao Delphi
- TsgcFMXGrid
- Demos do .NET

Classes do .NET

Todas as classes estão no namespace **esegece.sgcGrid**. Os três controles de grade são hosts finos sobre o modelo e o renderizador, neutros em relação ao framework, que são compartilhados pelos hosts, pela saída headless e pelas pontes.

Classes de host

Host	Classe	Classe base	Delphi
WinForms	TsgcWinFormsGrid	TsgcUIWinFormsControl	TsgcVCLGrid
WPF	TsgcWpfGrid	TsgcUIWpfControl	TsgcFMXGrid
Avalonia	TsgcAvaloniaGrid	TsgcUIAvaloniaControl	TsgcFMXGrid

As três grades implementam **IsgcUITable**, de modo que um gráfico do sgcCharts pode lê-las por meio do **TsgcChartDataSource_Table**.

Tipos neutros em relação ao framework

Tipo	Descrição
TsgcGridColumns, TsgcGridColumn	As colunas: **FieldName**, **Caption**, **DataType**, **Width**, **MinWidth**, **Alignment**, **DisplayFormat**, **Visible**, **Editable**, **SortOrder**, **SortIndex**, **PickList**, além de **GroupIndex**, **SummaryType** e **SummaryFormat** para agrupamento e resumos. Veja TsgcGridColumn.
TsgcGridData	O armazenamento de linhas indexado por chave e seguro para threads: **AppendRow**, **UpdateRow**, **UpdateCell**, **DeleteRow**, **Clear**, **BeginUpdate** e **EndUpdate**, carimbos de alteração, modo tail com **MaxRows**, ordenação, **FilterText** e **OnFilterRow**, modo virtual (**Virtual**, **VirtualRowCount**, **OnGetValue**) e **FormatSettings**. **LayoutVersion** avança a cada mudança que possa alterar o agrupamento ou a árvore, e **SetLayoutFields** adiciona campos cujas mudanças de valor também a fazem avançar. Veja TsgcGridData.
TsgcGridDataType	`gdtText` , `gdtInteger` , `gdtFloat` , `gdtCurrency` , `gdtDateTime` e `gdtBoolean` .
TsgcGridFormatRules, TsgcGridFormatRule	Formatação condicional: **FieldName**, **Operator** (de `gfoEqual` a `gfoIsEmpty`), **Value**, **Value2**, **BackColor**, **ForeColor**, **Bold**, **ApplyToRow**, **Enabled** e **StopOnMatch**. Veja TsgcGridFormatRule.
TsgcGridSummaryType	`gsuNone` , `gsuSum` , `gsuCount` , `gsuAverage` , `gsuMin` e `gsuMax` .
TsgcGridLineView, TsgcGridGroupInfo	As linhas de exibição de um TsgcGridData: linhas de grupo, linhas de dados, linhas de árvore e faixas de detalhe, com **GroupBy**, **GroupCount**, **GroupInfo**, **GroupSummary** e o estado expandido. Veja Agrupamento e Resumos do .NET.
TsgcGridTreeOptions	Linhas de árvore: **KeyField**, **ParentField**, **TreeField**, **Indent** e **Expanded**. Veja Linhas de Árvore e Mestre Detalhe do .NET. Somente na edição Registered.

TsgcGridDetailOptions	Faixas de detalhe: **Active**, **Source**, **MasterKeyField**, **DetailKeyField**, **Lines** e **Columns**. Veja Linhas de Árvore e Mestre Detalhe do .NET. Somente na edição Registered.
TsgcGridDataStore	Um contêiner não visual de linhas de detalhe, preenchido por código ou por um **TsgcGridSource_DataSet**. Somente na edição Registered.
TsgcGridXLSXWriter	Grava um TsgcGridData e sua visão em linhas como uma pasta de trabalho do Excel. Veja Exportação para Excel do .NET. Somente na edição Registered.
TsgcGridRenderer, TsgcGridRenderOptions	Desenha um quadro de um **TsgcGridData** em um **TsgcUICanvas** e o grava como SVG ou, com o esegece.sgcGrid.Rendering, como PNG. Veja Renderização Headless no .NET.
sgcGrid_IO	Importação e exportação de JSON e CSV, veja Vinculação de DataTable, JSON e CSV no .NET.
TsgcGridSource_DataSet	Vincula um `System.Data.DataTable` a uma grade, veja Vinculação de DataTable, JSON e CSV no .NET.
TsgcGridFeedSource e descendentes	As fontes de feed em tempo real do esegece.sgcGrid.sgcWebSockets, veja Feeds em Tempo Real no .NET. Somente na edição Registered.
TsgcGridAIQuery	Perguntas em linguagem natural do esegece.sgcGrid.sgcHTML, veja Pergunte à Sua Grade no .NET. Somente na edição Registered.

Grade

Os mesmos membros em **TsgcWinFormsGrid**, **TsgcWpfGrid** e **TsgcAvaloniaGrid**.

Membro	Tipo	Descrição
Columns	TsgcGridColumns	As colunas.
Data	TsgcGridData	O armazenamento de linhas, somente leitura. Pode ser gravado a partir de qualquer thread.
FormatRules	TsgcGridFormatRules	As regras de formatação condicional.
Renderer	TsgcGridRenderer	O renderizador da grade, somente leitura.
HeaderHeight, RowHeight	int	Altura do cabeçalho e de uma linha, 32 e 28 por padrão.
GridLinesH, GridLinesV, AlternateRows	bool	Linhas de grade horizontais e verticais e o tom das linhas alternadas, true por padrão.
SelectionMode	TsgcGridSelectionMode	`gsmRow` (padrão) ou `gsmCell`.
Theme, AccentColor	TsgcUITheme, uint	Tema da grade e um destaque opcional `$AARRGGBB`, 0 para o destaque do tema.
FontSize	double	Tamanho da fonte das células e do cabeçalho, 13 por padrão.

MaxRows, TailMode	int, bool	Modo tail: as linhas mais antigas são descartadas além de **MaxRows**, e **TailMode** mantém a última linha à vista.
FilterText	string	Mostra apenas as linhas cujo texto formatado o contém.
FlashChanges, FlashDuration, FlashMode	bool, int, TsgcGridFlashMode	Flash de alterações, 800 ms por padrão, no fundo ou no texto.
FlashUpColor, FlashDownColor, FlashChangedColor	uint	Cores do flash, 0 para as cores do tema.
ReadOnly	bool	True por padrão. False habilita a edição no local das colunas **Editable**. Somente na edição Registered.
SelectedRow, SelectedCol, SelectedKey	int, int, string	A seleção. A seleção acompanha sua linha pela chave quando as linhas são reordenadas.
TopRow, ScrollX, ScrollToRow	int, int, method	A posição de rolagem. **TopRow** conta linhas de exibição, que são as linhas da visão quando nada está agrupado.
RowAt, ColAt	method	A linha da exibição e a coluna sob um ponto.
StartEdit, CommitEdit, CancelEdit, EditorActive	method, bool	Edição no local a partir do código. Somente na edição Registered.
SaveToSVG, SaveToPNG	method	Grava a exibição atual como imagem.
OnCellClick, OnCellDblClick	TsgcGridCellEvent	Uma célula foi clicada, com a linha da exibição e o índice da coluna.
OnSelectionChange, OnSortChange	EventHandler	A seleção ou a ordenação mudou.
OnColumnResized	TsgcGridColumnEvent	O usuário redimensionou uma coluna.
OnGetCellStyle	TsgcGridCellStyleEvent	Altera as cores de uma célula antes de ela ser desenhada.
OnBeforeEdit, OnValidate, OnEdited, OnEditError	events	Edição no local, veja Edição no Local no .NET. Somente na edição Registered.
ShowGroupPanel, GroupPanelHeight, GroupIndent, GroupsExpanded, GroupSummaries	bool, int, int, bool, bool	Opções de exibição do agrupamento, veja Agrupamento e Resumos do .NET.
ShowFooter, FrozenColumns	bool, int	O rodapé de resumos e as colunas mantidas à vista enquanto as outras rolam.
Lines, FocusedLine	TsgcGridLineView, int	As linhas de exibição e a linha com o foco.
GroupBy, ClearGrouping, ExpandAll, CollapseAll, ExpandGroup, CollapseGroup, GroupCount, GroupInfo, FooterSummary, FooterSummaryText	method	Agrupamento e resumos por código.

OnGroupingChange, OnGroupExpandChange, OnGetGroupText	eventos	O agrupamento ou o estado de um grupo mudou, e o título de uma linha de grupo.
Tree, Detail	TsgcGridTreeOptions, TsgcGridDetailOptions	Linhas de árvore e faixas de detalhe, veja Linhas de Árvore e Mestre Detalhe do .NET. Somente na edição Registered.
ExpandNode, CollapseNode, ExpandDetail, CollapseDetail, DetailExpanded, DetailData	method	Nós de árvore e faixas de detalhe por código. Somente na edição Registered.
OnTreeExpandChange, OnDetailExpandChange, OnDetailCellClick, OnGetDetailData	eventos	Eventos de árvore e de detalhe. Somente na edição Registered.
SaveToXLSX	method	Grava a visão como uma pasta de trabalho do Excel, veja Exportação para Excel do .NET. Somente na edição Registered.
VertBarVisible, HorzBarVisible, BorderStyle	bool, bool, TsgcGridBorderStyle	WPF e Avalonia: as barras de rolagem desenhadas e a borda. O WinForms usa o **BorderStyle** de `System.Windows.Forms` e barras de rolagem nativas.

Veja também

- Visão Geral do .NET
- Diferenças do .NET em Relação ao Delphi
- Renderização Headless no .NET
- Vinculação de DataTable, JSON e CSV no .NET
- Feeds em Tempo Real no .NET
- Pergunte à Sua Grade no .NET
- Agrupamento e Resumos do .NET
- Linhas de Árvore e Mestre Detalhe do .NET
- Exportação para Excel do .NET
- TsgcGridData
- TsgcGridRenderer

Diferenças do .NET em Relação ao Delphi

A porta .NET mantém os nomes, os valores padrão e as regras de renderização do Delphi; sua saída SVG é idêntica byte a byte à do Delphi. As diferenças abaixo vêm dos frameworks dos hosts e da biblioteca de classes do .NET.

Data e FormatRules em vez de IsgcGridDataHost

No Delphi, uma fonte encontra os dados de uma grade pela interface IsgcGridDataHost e IsgcUITable. A porta .NET não tem essa interface. **TsgcWinFormsGrid**, **TsgcWpfGrid** e **TsgcAvaloniaGrid** expõem **Data** e **FormatRules** como propriedades públicas, e **TsgcGridSource_DataSet**, as fontes de feed e o **TsgcGridAIQuery** as encontram por reflexão: **Grid** aceita qualquer objeto com uma propriedade pública **Data** do tipo **TsgcGridData**, e a consulta de IA também lê uma propriedade pública **FormatRules** do tipo **TsgcGridFormatRules** para seus destaques. Assim, sua própria classe pode ser o **Grid** de uma fonte, veja o exemplo em Pergunte à Sua Grade no .NET. **GridData**, quando atribuído, sempre prevalece sobre **Grid**.
Não existe notificação de liberação no .NET. Quando **Grid** é um `IComponent`, como um controle WinForms, a fonte escuta o evento `Disposed` dele e então limpa **Grid**. Para qualquer outro objeto, defina **Grid** como null antes de a grade desaparecer. **GridData** não é monitorado.

Barras de rolagem

O **TsgcWinFormsGrid** usa as barras de rolagem nativas do Windows, como a grade VCL. **TsgcWpfGrid** e **TsgcAvaloniaGrid** desenham suas próprias barras de rolagem, como a grade FireMonkey, com arraste do thumb e paginação pela trilha; **VertBarVisible** e **HorzBarVisible** as mostram ou ocultam.

FontSize no WPF, Theme no Avalonia

A grade desenha seu próprio texto, então o **TsgcWpfGrid** declara seu próprio **FontSize** com `new`, o que oculta `Control.FontSize`. Por meio de uma referência do tipo `Control`, ele é a propriedade do WPF, que a grade não lê. Da mesma forma, todo controle Avalonia já tem uma propriedade **Theme**, o `ControlTheme` de `StyledElement`: o **TsgcAvaloniaGrid** declara seu próprio **Theme** do tipo **TsgcUITheme** com `new`. Por meio de uma referência do tipo da grade, ele é o tema sgc; por meio de uma referência do tipo `Control` ou `StyledElement`, é o tema de controle do Avalonia.
No WPF e no Avalonia, as duas propriedades aceitam vinculações e styles, como a seleção e as demais opções de exibição, veja Vinculação de dados no .NET.

Toque

No Avalonia, um contato de toque move a grade, o conteúdo acompanha o dedo sem rolagem cinética, como faz o gesto de pan do FireMonkey, e um toque seleciona a célula. Lá um contato de toque nunca arrasta um cabeçalho para o painel de agrupamento; um toque em um chevron, em um expansor ou em uma faixa de detalhe age como um clique. O WPF não tem pan por toque: toque e caneta são promovidos a entrada de mouse, então um toque seleciona e as barras de rolagem desenhadas rolam. O WinForms usa o mouse e a roda.

Tipos de dados

Os tipos do CLR são mapeados para os tipos de dados da grade como os tipos de campo do Delphi: `byte`, `short`, `int`, `long` e os demais tipos inteiros são mapeados para `gdtInteger`; `float`, `double` e `decimal` são mapeados para `gdtFloat`, como os campos `ftBCD` e `ftFMTBcd` do Delphi; `DateTime`, `DateTimeOffset` e `TimeSpan` são mapeados para `gdtDateTime`; `bool` é mapeado para `gdtBoolean`; todo o resto é `gdtText`. O CLR não tem um tipo currency, portanto `gdtCurrency` só vem de uma coluna que você mesmo adiciona.

DataTable em vez de TDataSet

O **TsgcGridSource_DataSet** vincula um `System.Data.DataTable`, o equivalente .NET de um `TDataSet` do Delphi. Um DataTable está sempre aberto e não tem cursor, portanto não há registro atual nem bookmark para restaurar depois de um carregamento. Os eventos de **LiveUpdates** (`RowChanged`, `RowDeleted`, `TableCleared`) fazem o papel do data link do Delphi, e as edições não concluídas e o `AcceptChanges` são ignorados. Veja Vinculação de DataTable, JSON e CSV no .NET.

Regras de formatação sem ID de item

As regras de formatação do .NET não têm ID de item de coleção. **TsgcGridAIQuery.Reset** remove as regras que adicionou por referência, e não por ID; as colunas continuam sendo encontradas pelo seu **ID**.

Edição das linhas de árvore, das faixas de detalhe e da exportação para Excel

No Delphi, as linhas de árvore, as faixas de detalhe e **SaveToXLSX** são compiladas sob a definição `SGC_GRID_ADVANCED`. Os pacotes Community do .NET não contêm **TsgcGridTreeOptions**, **TsgcGridDetailOptions**, **TsgcGridDataStore**, **TsgcGridXLSXWriter** nem os membros da grade que os usam; agrupamento, resumos, rodapé e colunas congeladas estão nas duas edições. Veja Linhas de Árvore e Mestre Detalhe do .NET e Exportação para Excel do .NET.

Eventos de agrupamento, árvore e detalhe

Os novos eventos da grade são eventos C#, como os demais eventos das grades. Seus delegates, **TsgcGridGroupExpandEvent**, **TsgcGridRowExpandEvent** e **TsgcGridDetailCellEvent**, são declarados uma única vez em esegece.sgcGrid e compartilhados pelos três hosts. Métodos sem argumentos, como **GroupCount**, continuam sendo métodos.

Origem do detalhe

Detail.Source é um `object`: um **TsgcGridDataStore**, um **TsgcGridData** ou qualquer objeto com uma propriedade pública **Data** do tipo **TsgcGridData**, encontrado como as origens encontram uma grade. Sem uma notificação de liberação, uma Source que é um `IComponent` é limpa quando seu evento `Disposed` é disparado.

Arquivos do Excel

A pasta de trabalho é compactada com `System.IO.Compression.DeflateStream`. Suas partes XML são os mesmos bytes da saída do Delphi para os mesmos dados; apenas os bytes compactados, e portanto o tamanho do arquivo, diferem.

Serialização no designer das novas opções

Não há DFM. No WinForms as sete opções de agrupamento têm valores padrão e são serializadas somente quando diferem, e **Tree** e **Detail** são serializadas como conteúdo. No WPF e no Avalonia as opções são propriedades de dependência e propriedades com estilo.

Sem verbos de designer

As grades do Delphi adicionam verbos de tempo de design à IDE. As grades .NET não têm nenhum: adicione as **Columns** e as **FormatRules** em código, como fazem os exemplos de início rápido.

Veja também

- Visão Geral do .NET
- Classes do .NET
- Início Rápido WPF
- Início Rápido Avalonia
- Vinculação de DataTable, JSON e CSV no .NET
- Pergunte à Sua Grade no .NET
- Linhas de Árvore e Mestre Detalhe do .NET
- Exportação para Excel do .NET
- IsgcGridDataHost e IsgcUITable

Vinculação de dados no .NET

Este tópico trata somente de **TsgcWpfGrid** e **TsgcAvaloniaGrid**. Suas propriedades escalares visíveis ao usuário são propriedades de dependência do WPF e propriedades estilizadas do Avalonia, portanto aceitam vinculações, styles e setters XAML, e um view model pode controlar a grade sem code-behind. **TsgcWinFormsGrid** não muda: suas propriedades são propriedades CLR simples, como antes.

Propriedades vinculáveis

As propriedades abaixo aceitam uma vinculação nas duas grades. As marcadas com TwoWay são vinculadas em TwoWay por padrão, de modo que uma vinculação sem `Mode` também leva de volta ao view model as alterações da grade. Todas as outras são vinculadas em OneWay por padrão.

Propriedade	Tipo	Modo padrão
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 e **SelectedCol** acompanham um clique, uma tecla ou o código. No Avalonia são propriedades diretas. No WPF, **FontSize** é declarada com `new` e oculta `Control.FontSize`, e no Avalonia **Theme** oculta o tema de controle, veja Diferenças do .NET em relação ao Delphi. Vincule-as no próprio elemento da grade, como fazem os exemplos.

Comportamento

- O modelo da grade mantém a sua validação. Quando ele rejeita ou corrige um valor, por exemplo uma **FlashDuration** negativa ou um **SelectedRow** após a última linha, a grade grava de volta na propriedade o valor que manteve. No WPF, quando o valor veio de uma origem TwoWay, o valor mantido também volta a essa origem um tick do dispatcher depois, quando a vinculação terminou a sua transferência.
- Quando o código altera o modelo diretamente por meio de **Renderer.Options**, o novo valor é copiado de volta para a propriedade vinculável somente na thread da interface, de modo que um view model vinculado nunca é atualizado a partir de outra thread.
- **OnSelectionChange** e os demais eventos disparam para toda alteração, venha ela de uma vinculação, de um style, do código, do mouse ou do teclado.
- A seleção acompanha a sua chave de linha quando **FilterText** move a linha, e o **SelectedRow** vinculado também a acompanha.

Propriedades que continuam CLR

- **Data**, **Columns** e **FormatRules** continuam sendo propriedades CLR, porque **TsgcGridSource_DataSet**, as origens de feed e **TsgcGridAIQuery** as encontram por reflexão. Preencha-as pelo código ou por uma origem, veja Vinculação de DataTable, JSON e CSV no .NET e Feeds em Tempo Real no .NET.
- **TopRow** e **ScrollX** continuam sendo propriedades CLR, porque são estado da exibição que muda a cada passo de rolagem.
- As propriedades de estado somente leitura, como **SelectedKey**, **VertBarVisible**, **HorzBarVisible** e **Renderer**, continuam sendo propriedades CLR, porque não há nada para vincular a elas.

View model

Os dois exemplos usam o mesmo view model. **SelectedRow** é vinculado em TwoWay, **FilterText** é compartilhado com uma caixa de texto, e **Theme** é calculado a partir de uma caixa de seleção.
MainViewModel.cs:

Exemplo WPF

MainWindow.xaml:
MainWindow.xaml.cs:

Exemplo Avalonia

A janela Avalonia usa vinculações compiladas, portanto define `x:DataType` como o view model.
MainWindow.axaml:
MainWindow.axaml.cs:
Definir `vm.SelectedRow = 3` seleciona a linha AMZN, e um clique em outra linha grava o seu índice de volta em `vm.SelectedRow`. Digitar na caixa de texto filtra a grade, e a caixa de seleção alterna o tema.

Veja Também

- Início Rápido WPF
- Início Rápido Avalonia
- Classes do .NET

- Diferenças do .NET em relação ao Delphi
- Vinculação de DataTable, JSON e CSV no .NET
- Feeds em Tempo Real no .NET

Renderização Headless no .NET

O renderizador não precisa de controle nem de janela, então um serviço, uma aplicação web ou um servidor Linux pode gravar a imagem de uma grade. **TsgcGridRenderer.SaveToSVG** grava SVG a partir do pacote core. **SaveToPNG**, um método de extensão do esegece.sgcGrid.Rendering, grava PNG por meio do SkiaSharp.

Exemplo

Este programa de console cria o modelo, ordena, adiciona uma regra de formatação e grava o mesmo quadro como SVG e como PNG com o dobro do tamanho:
O programa grava:

```
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
```

Observações

- Largura e altura são em pixels independentes de dispositivo; o último argumento de **SaveToSVG** e **SaveToPNG** é a escala, então `520 x 170` com escala 2 grava uma imagem de 1040 x 340 pixels.
- `sgcGrid_Render.sgcGridRenderState()` retorna um estado sem seleção e sem rolagem. Defina seus campos para renderizar um quadro rolado ou com seleção.
- As opções de renderização (**Theme**, **FontSize**, **RowHeight**, as configurações de flash e **FormatRules**) são as mesmas que os controles de grade expõem, e um controle de grade grava sua própria exibição com **SaveToSVG** e **SaveToPNG**.
- O texto SVG é idêntico byte a byte ao que o renderizador do Delphi grava para os mesmos dados. A saída PNG usa o antialiasing do Skia, portanto pode diferir do raster do Delphi em pixels isolados.
- **UseInvariantFormat** torna **CellText** independente da cultura da máquina.

Veja também

- Visão Geral do .NET
- Classes do .NET
- Desempenho no .NET
- Renderização Headless
- TsgcGridRenderer

Vinculação de DataTable, JSON e CSV no .NET

O **TsgcGridSource_DataSet** vincula um `System.Data.DataTable` a uma grade, e a classe **sgcGrid_IO** carrega e salva as linhas como JSON e CSV. Ambos ficam no pacote core e funcionam com todos os hosts e em modo headless.

Vinculação de DataTable

- Defina **Grid** como o controle de grade, ou **GridData** como o seu **Data**.
- Com **AutoColumns** true (o padrão) e uma grade sem colunas, as colunas são criadas a partir do DataTable: o nome do campo é `ColumnName`, o título é `Caption`, uma coluna com `ColumnMapping Hidden` fica oculta, e o tipo de dados segue o tipo CLR (`decimal` é mapeado para `gdtFloat`).
- **KeyField** indica a coluna cujo valor é a chave da linha. Com uma chave, um recarregamento atualiza as linhas no local e mantém o flash de alteração, acrescenta as chaves novas e exclui as chaves que desapareceram. Sem chave, cada carregamento substitui as linhas.
- Com **LiveUpdates** true (o padrão), uma alteração confirmada, uma exclusão ou uma limpeza do DataTable chega à grade imediatamente. **Load** carrega a qualquer momento e **Refresh** recarrega.

O programa grava:

```
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
```

Atribuir null a **DataSet** desvincula a tabela e mantém as linhas. Descartar o DataTable tem o mesmo efeito.

JSON e CSV

GridLoadFromJSON lê um array de objetos; um nome de campo com pontos, como `weather.temp`, lê um objeto aninhado, e **GridSaveToJSON** o grava de volta como tal. **GridLoadFromCSV** e **GridSaveToCSV** usam as aspas da RFC 4180, qualquer delimitador e uma linha de cabeçalho. O argumento do campo chave indica a chave da linha e, com `aAutoColumns`, as colunas são criadas a partir do primeiro objeto ou da linha de cabeçalho, com os tipos deduzidos dos valores. As sobrecargas de string, stream e arquivo leem UTF-8 e ignoram um BOM, e gravam UTF-8 sem ele.
O programa grava:

```
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
```

Os números do CSV usam a cultura invariante, a menos que `aInvariant` seja false; nesse caso são usados os **FormatSettings** dos dados. As linhas são gravadas na ordem da exibição, e `aVisibleOnly` grava somente as colunas visíveis.

Veja também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em Relação ao Delphi

- TsgcGridSource_DataSet
- Vinculação de DataSet
- JSON e CSV

Edição no Local no .NET

Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.
As três grades editam uma célula no local quando **ReadOnly** é false e a coluna é **Editable**. Uma caixa de texto se abre sobre a célula, ou uma caixa de combinação quando a coluna tem uma **PickList**. Uma célula booleana alterna com Espaço ou com um clique. A edição começa com F2, Enter, um caractere digitado, um clique duplo ou **StartEdit**, é confirmada com Enter, Tab ou quando o editor perde o foco, e é cancelada com Esc ou **CancelEdit**.

Eventos

Evento	Descrição
OnBeforeEdit	Antes de o editor abrir. Defina `aAllow` como false para vetar a edição.
OnValidate	Antes de o valor ser gravado, já convertido para o **DataType** da coluna. Altere `aValue`, ou defina `aAccept` como false para rejeitá-lo; o editor então permanece aberto.
OnEdited	Depois de o valor ser gravado com **Data.UpdateCell**, com o valor antigo e o novo.
OnEditError	O texto não pode ser convertido para o **DataType** da coluna. Altere `aMessage` para definir a mensagem.

O argumento de coluna dos eventos de edição é o índice em **Columns**, e não a posição visível.

Exemplo

Um feed em tempo real nunca sobrescreve um editor aberto: o editor acompanha sua linha pela chave quando as linhas são reordenadas e fecha quando a linha é excluída. No WinForms os editores são os controles filhos **TsgcWinFormsGridEdit** e **TsgcWinFormsGridCombo**; no WPF e no Avalonia são um `TextBox` e um `ComboBox` hospedados. Quando um mesmo projeto compila as duas edições, proteja o código de edição com `#if !SGC_COMMUNITY`.

Veja também

- Visão Geral do .NET
- Classes do .NET
- Início Rápido WinForms
- Edição no Local
- Seleção e Teclado
- Edições

Feeds em Tempo Real no .NET

Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.
O pacote esegece.sgcGrid.sgcWebSockets alimenta uma grade a partir de conexões do sgcWebSockets .NET. **TsgcGridFeedSource** é a classe base; **TsgcGridFeedSource_WebSocket** lê um **TsgcWebSocketClient**, **TsgcGridFeedSource_MQTT** lê as publicações de um cliente MQTT e **TsgcGridFeedSource_SSE** lê server-sent events. Cada mensagem é lida como JSON e mapeada para linhas da grade.

Mapeamento

Propriedade	Descrição
Grid, GridData	O destino: um controle de grade, encontrado por meio de sua propriedade pública **Data**, ou um **TsgcGridData**. **GridData** prevalece.
RowsPath	Caminho das linhas na mensagem: um array de objetos de linha ou um único objeto de linha. Vazio quando a própria mensagem é a linha ou o array.
KeyPath	Caminho da chave da linha dentro de um objeto de linha, por exemplo `symbol` ou `id.value`. Uma linha sem chave é ignorada e informada por meio de **OnError**.
Fields	Um `FieldName=json.path` por linha, por exemplo `price=quote.last` ou `bid=book.bid[0]`. Uma linha sem `=` lê o caminho de mesmo nome. Quando **Fields** está vazio, cada coluna é lida do caminho do seu **FieldName**. Somente os caminhos presentes em uma linha são gravados, de modo que uma mensagem pode trazer uma atualização parcial.
DeletePath	Um caminho booleano no objeto de linha: true exclui a linha da chave.
MaxRowsPerMessage	Limita as linhas aplicadas a partir de uma mensagem, 0 para sem limite. As linhas excedentes são informadas por meio de **OnError**.
Active	False ignora todas as mensagens.
OnRow, OnError	Uma linha está prestes a ser aplicada (defina `aAccept` como false para ignorá-la), ou uma mensagem não pôde ser aplicada.

Os números JSON são gravados como double, os booleanos como bool, as strings como string e o null como o null da grade.

Fontes

Classe	Propriedade do cliente	Lê
TsgcGridFeedSource_WebSocket	**Client** (`TsgcWebSocketClient`)	Todas as mensagens de texto.
TsgcGridFeedSource_MQTT	**Client** (`TsgcWSProtocol_MQTT_Client`)	As publicações cujo tópico corresponde a **Topic**, com os curingas `+` e `#`.

TsgcGridFeedSource_SSE	**Client** (`TsgcHTTP1Client`) ou **SSE-Client** (`TsgcSSEClient`)	Os dados de cada server-sent event. **SSEClient** é um acréscimo do .NET.

As fontes se inscrevem nos eventos do cliente ao lado dos manipuladores da aplicação e nunca os substituem.

Exemplo

Este programa de console inicia um **TsgcWebSocketServer** no mesmo processo, na interface de loopback, e alimenta um modelo de grade com três mensagens: um lote de linhas, uma atualização parcial com uma exclusão e um único objeto de linha.
O programa grava:

```
AAPL 228.05 x 300
NVDA 121.40 x 900
JPM 212.80 x 50
```

Threads

Todas as linhas de uma mensagem são aplicadas dentro de um único **BeginUpdate** e **EndUpdate**, na thread em que a mensagem chega. **TsgcGridData** é seguro para threads e as grades encaminham o redesenho para a thread da interface, portanto não é necessária nenhuma sincronização. **OnRow** e **OnError** também são executados nessa thread: encaminhe para a thread da interface antes de tocar em um controle lá.

Veja também

- Visão Geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Demos do .NET
- TsgcGridFeedSource
- TsgcGridFeedSource_WebSocket
- TsgcGridFeedSource_MQTT
- TsgcGridFeedSource_SSE
- Feeds em Tempo Real

Pergunte à Sua Grade no .NET

Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.
O **TsgcGridAIQuery** do pacote esegece.sgcGrid.sgcHTML transforma uma pergunta em linguagem natural em filtros, ordenações e destaques de uma grade. A pergunta é respondida pelo **TsgcHTMLAIAssistant** do esegece.sgcHTML: por um modelo de linguagem quando o assistente tem um provedor, e por seu analisador baseado em regras quando **Assistant** é null ou não tem provedor. O analisador baseado em regras não precisa de rede e entende perguntas como `price greater than 100`, `sort by volume descending` e `highlight change less than 0`.

Como a resposta é aplicada

- A resposta é verificada em relação às colunas da grade (**FieldName**, **Caption** e **DataType**) antes de qualquer coisa ser aplicada.
- Os filtros são combinados com AND por meio de **Data.OnFilterRow**. Um manipulador que você atribuiu é mantido e chamado primeiro, e uma linha que ele rejeita continua rejeitada.
- As ordenações substituem o **SortOrder** e o **SortIndex** das colunas.
- Os destaques são adicionados às **FormatRules** da grade com **ApplyToRow**, nas cores do estilo do assistente.
- Agrupamento e limites de linhas não são suportados pela grade; são informados em **LastExplanation**.
- Cada **Ask** desfaz primeiro o anterior. **Reset** restaura o manipulador de filtro e a ordenação de colunas anteriores e remove apenas as regras que a consulta adicionou.

Ask é síncrono e deve ser chamado a partir da thread da interface, pois altera as colunas e as regras de formatação. Com um modelo de linguagem, ele bloqueia enquanto o provedor responde. **OnApplied** e **OnError** informam o resultado.

Exemplo

O **Grid** da consulta é qualquer objeto com uma propriedade pública **Data**; para os destaques ele também precisa de uma propriedade pública **FormatRules**. Os controles de grade têm as duas. Este programa de console usa uma pequena classe própria e nenhum assistente, portanto o analisador baseado em regras responde:
O programa grava:

```
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 um modelo de linguagem, atribua a **Assistant** um **TsgcHTMLAIAssistant** com um provedor configurado.

Veja também

- Visão Geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Diferenças do .NET em Relação ao Delphi
- Demos do .NET
- TsgcGridAIQuery
- Pergunte à Sua Grade

Desempenho no .NET

A grade desenha apenas as linhas e colunas visíveis e é redesenhada no máximo **MaxFPS** vezes por segundo, por mais rápido que os dados mudem. O host WinForms é medido por um gate de desempenho na suíte de testes, uma porta do teste de desempenho do Delphi com o caminho de desenho do host e um feed em tempo real acrescentados.

A cena

- Uma área de cliente de `1920 x 1080` de um **TsgcWinFormsGrid**, 12 colunas de 160 pixels independentes de dispositivo (texto, inteiro, float, currency, hora e booleano) e uma altura de linha de 23.
- 100.000 linhas indexadas por chave.
- Uma thread de trabalho envia 1000 atualizações por segundo, metade delas para as linhas próximas ao topo da exibição.

Tempo de quadro

O tempo de quadro é o desenho completo do controle, renderização e blit, ao longo de 60 quadros enquanto o feed está em execução. O orçamento é de 16 ms por quadro em um processo de 64 bits, um quadro a 60 fps, e de 25 ms em um processo de 32 bits.

Runtime	Tempo de quadro
.NET 8, antes das máscaras de cobertura de glifos do sgcUICore	44 ms
.NET 8	11,4 a 13,1 ms
.NET Framework 4.7.2	7,3 a 7,7 ms

A primeira linha é a medição que levou às máscaras de cobertura de glifos do sgcUICore: cada texto de célula preenchia todo o seu contorno por meio do Skia.

Ritmo

A segunda fase do gate mede a taxa de redesenho sustentada com **MaxFPS** em 60 sobre um loop de mensagens real durante 3 segundos, e verifica se o feed mantém pelo menos 90 por cento da sua taxa e se o redesenho nunca excede **MaxFPS**. As faixas abaixo vêm de três execuções do gate.

Runtime	Taxa de redesenho
.NET 8	59,5 a 59,7 fps
.NET Framework 4.7.2	59,8 a 60 fps

No .NET 8, o percentil 95 do intervalo entre dois quadros é de 21,4 a 22,2 ms.

Executando o gate

O gate é sensível ao tempo e precisa de uma máquina de outra forma ociosa, por isso só é executado sob demanda: defina a variável de ambiente `SGC_GRID_RUN_PERF=1` e execute a categoria `Perf` dos testes WinForms.
Os números são acrescentados a `sgcGridQA_perf.txt`, ao lado do assembly de testes. O gate também grava, como referência, os tempos de **SaveToPNG** e de **Render** headless do mesmo quadro. O **Render** headless deste quadro leva 5,0 ms.

Veja também

- Visão Geral do .NET
- Início Rápido WinForms
- Renderização Headless no .NET
- Desempenho
- Atualizações em Tempo Real e Threads

Agrupamento e Resumos do .NET

As linhas podem ser agrupadas por uma ou mais colunas. Cada grupo recebe uma linha de grupo com um chevron de expansão, o texto do membro, a contagem de linhas e os resumos das colunas; um rodapé mostra os mesmos resumos sobre todas as linhas, e as colunas congeladas permanecem à vista enquanto as outras rolam na horizontal. Agrupamento, resumos, rodapé e colunas congeladas fazem parte das duas edições, em **TsgcWinFormsGrid**, **TsgcWpfGrid** e **TsgcAvaloniaGrid**.
O agrupamento é estado da visão. **TsgcGridData** mantém suas linhas planas e sua ordenação, e **IsgcUITable** continua informando as linhas da visão sem agrupar, de modo que um gráfico sgcCharts vinculado à grade não vê nenhuma mudança. A grade distribui as linhas por meio de um **TsgcGridLineView**, disponível em sua propriedade **Lines**.

Propriedades de coluna

Propriedade	Padrão	Descrição
GroupIndex	-1	Um valor de 0 ou mais agrupa as linhas pela coluna nesse nível. Os níveis são ordenados por **GroupIndex** e depois pelo índice da coluna. **GroupBy**, **ClearGrouping** e o painel de agrupamento o definem para você.
SummaryType	`gsuNone`	`gsuSum`, `gsuCount`, `gsuAverage`, `gsuMin` ou `gsuMax`. O resumo é mostrado em cada linha de grupo e no rodapé.
SummaryFormat	vazio	O formato do resumo, como em **DisplayFormat**: por exemplo `#,##0.00`, com texto literal entre aspas duplas, `"Total: "#,##0.00`. Vazio usa **DisplayFormat**; uma contagem usa `0`.

Alterar essas propriedades dispara apenas o **OnChange** das colunas e nunca recarrega os dados. Os valores do resumo são `null` quando não há valor, um `int` para uma contagem, um `double` para uma soma, uma média, um mínimo ou um máximo, e um `DateTime` para o mínimo ou o máximo de uma coluna de data.

Membros da grade

Os mesmos membros existem em **TsgcWinFormsGrid**, **TsgcWpfGrid** e **TsgcAvaloniaGrid**. No WPF e no Avalonia as sete opções são propriedades de dependência e propriedades com estilo, portanto podem ser definidas em XAML, vinculadas e estilizadas.

Membro	Tipo	Descrição
ShowGroupPanel, GroupPanelHeight	bool, int	O painel acima do cabeçalho: arraste para lá o cabeçalho de uma coluna para agrupar por ela. Oculto por padrão, com 36 pixels de altura.
GroupIndent	int	Recuo de cada nível de grupo, 20 por padrão.
GroupsExpanded	bool	O estado dos novos grupos, true por padrão.
GroupSummaries	bool	Mostra os resumos nas linhas de grupo, true por padrão.
ShowFooter	bool	Mostra o rodapé com os resumos sobre todas as linhas, false por padrão.
FrozenColumns	int	As primeiras colunas visíveis mantidas à vista enquanto as outras rolam, 0 por padrão.

GroupBy, ClearGrouping	method	`GroupBy("region", "category")` agrupa por esses campos nessa ordem e limpa qualquer outro **GroupIndex**; **ClearGrouping** mostra as linhas planas novamente.
ExpandAll, CollapseAll, ExpandGroup, CollapseGroup	method	Expandem ou recolhem todos os grupos, ou um grupo pelo seu índice.
GroupCount, GroupInfo	method	O número de grupos e o **TsgcGridGroupInfo** de um grupo.
FooterSummary, FooterSummaryText	method	O valor do rodapé de uma coluna (null quando não há nenhum) e seu texto como o rodapé o mostra.
Lines	TsgcGridLineView	A visão em linhas da grade: linhas de grupo, linhas de dados e, na edição Registered, linhas de árvore e faixas de detalhe.
FocusedLine	int	A linha com o foco. Em uma linha de grupo **SelectedRow** é -1 e **FocusedLine** informa a linha.
OnGroupingChange	EventHandler	As colunas agrupadas ou sua ordem mudaram, por código ou pelo painel de agrupamento.
OnGroupExpandChange	TsgcGridGroupExpandEvent	Um grupo foi expandido ou recolhido. `aGroup` é -1 depois de **ExpandAll** e **CollapseAll**.
OnGetGroupText	TsgcGridGetGroupTextEvent	Altera o título de uma linha de grupo antes de ela ser desenhada.

Os índices de linha continuam sendo linhas da visão: **SelectedRow**, **OnCellClick** e **Data** usam as mesmas linhas que sem agrupamento. **TopRow** conta linhas de exibição, que são as linhas da visão quando nada está agrupado.

Informações do grupo

Field	Descrição
Level	0 para o nível mais externo.
Parent	O grupo pai, -1 no nível 0.
Column	O índice em **Columns** da coluna agrupada.
RowCount	As linhas da visão no grupo, incluindo seus grupos filhos.
FirstRow	A linha da visão da primeira linha do grupo.
Text, Value, Blank	O texto do membro, seu valor tipado, e true para o membro em branco, mostrado como `(blank)`.
Expanded	O grupo está expandido.
Line	A linha da linha de grupo, -1 dentro de um grupo recolhido.
Path	Os textos dos membros a partir do nível 0.

Mouse e teclado

- Arraste o cabeçalho de uma coluna para o painel de agrupamento para agrupar por ela; um marcador mostra onde ela será solta. Arraste um chip ao longo do painel para mudar a ordem, arraste-o para fora para removê-lo, clique nele para inverter a ordenação do grupo. Esc cancela um arraste.
- Clique no chevron de uma linha de grupo, ou dê um clique duplo na linha, para expandi-la ou recolhê-la.
- Em uma linha de grupo, Right e Left expandem e recolhem, + e a tecla menos fazem o mesmo, * expande todos os grupos e Enter alterna o grupo. Ctrl+C em uma linha de grupo copia seu título.
- No Avalonia um contato de toque ainda move a grade; um toque em um chevron age como um clique.

Exemplo

Um formulário WinForms que agrupa pedidos por região e categoria, com uma soma e uma média em cada linha de grupo e no rodapé, duas colunas congeladas e um título personalizado para os grupos de região:
Os dados podem ser gravados de qualquer thread enquanto estão agrupados: a grade ressincroniza os grupos e os resumos na thread da interface e ritma o trabalho, de modo que a interface continua responsiva.

Agrupamento sem um controle

TsgcGridLineView funciona somente com um **TsgcGridData**, por exemplo em um serviço ou em um programa de console. **GroupSummary** retorna o valor do resumo de um grupo e de uma coluna, e **SetGroupExpanded** expande ou recolhe um grupo:
Saída:

```
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
```

Dados e uma visão em linhas também são o que **TsgcGridXLSXWriter** grava, veja Exportação para Excel do .NET.

Veja também

- Visão Geral do .NET
- Classes do .NET
- Linhas de Árvore e Mestre Detalhe do .NET
- Exportação para Excel do .NET
- Demos do .NET
- TsgcGridColumn
- Agrupamento
- Resumos
- Colunas Congeladas
- TsgcGridLineView

Linhas de Árvore e Mestre Detalhe do .NET

Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.
TsgcWinFormsGrid, **TsgcWpfGrid** e **TsgcAvaloniaGrid** mostram as linhas como uma árvore, construída a partir de um campo de chave e de um campo pai, e abrem uma faixa de detalhe sob uma linha mestre com suas linhas filhas vindas de uma segunda origem. O modo árvore e o agrupamento não ficam ativos ao mesmo tempo: no modo árvore o agrupamento é ignorado, o **GroupIndex** das colunas é mantido e o painel de agrupamento fica oculto.

Linhas de árvore

A propriedade **Tree** é um **TsgcGridTreeOptions**. A árvore fica ativa quando **KeyField** e **ParentField** estão ambos definidos.

Propriedade	Padrão	Descrição
KeyField	vazio	A coluna com a chave da linha.
ParentField	vazio	A coluna com a chave da linha pai. Linhas com chave de pai vazia, desconhecida ou igual à própria são raízes.
TreeField	vazio	A coluna que mostra o recuo e o botão de expandir. Vazio usa a primeira coluna visível.
Indent	16	Recuo de cada nível.
Expanded	false	O estado dos novos nós.

Os campos devem ser colunas da grade; podem estar ocultas. As chaves são comparadas como texto: inteiros e números de ponto flutuante sem fração usam seu texto inteiro, de modo que uma chave `int` e um pai `double` coincidem. Quando linhas compartilham uma chave, a primeira linha é dona dos filhos, e um ciclo é quebrado no primeiro nó repetido. Os irmãos mantêm a ordem da visão, de modo que ordenar ordena os irmãos, e com um filtro uma linha visível fica sob seu ancestral visível mais próximo. O estado expandido é mantido por chave, portanto sobrevive a um recarregamento.

Membro	Descrição
ExpandNode, CollapseNode	Expandem ou recolhem o nó de uma linha da visão.
ExpandAll, CollapseAll	No modo árvore expandem ou recolhem todos os nós.
OnTreeExpandChange	Um nó foi expandido ou recolhido; `aRow` é -1 depois de **ExpandAll** e **CollapseAll**.

Clique no botão de um nó para expandi-lo ou recolhê-lo. Em um nó, Right o expande ou vai para seu primeiro filho, e Left o recolhe ou vai para seu pai.

Mestre detalhe

A propriedade **Detail** é um **TsgcGridDetailOptions**. Uma coluna de expansão abre uma faixa sob uma linha mestre; a faixa é uma grade aninhada com seu próprio cabeçalho, posição de rolagem e seleção. Os detalhes se aninham em um único nível e são somente leitura.

Propriedade	Padrão	Descrição

Active	false	Mostra a coluna de expansão.
Source	null	As linhas de detalhe: um **TsgcGridDataStore**, um **TsgcGridData**, ou qualquer objeto com uma propriedade pública **Data** do tipo **TsgcGridData**, como outra grade.
MasterKeyField	vazio	A coluna mestre cujo valor é procurado nas linhas de detalhe.
DetailKeyField	vazio	A coluna da origem que contém a chave mestre.
Lines	8	Altura de uma faixa em linhas da grade, incluindo o cabeçalho aninhado.
Columns	vazio	As colunas das faixas. Vazio usa uma cópia das colunas da origem.

Membro	Descrição
ExpandDetail, CollapseDetail, DetailExpanded	Abrem, fecham e testam a faixa de uma linha da visão.
DetailData	O **TsgcGridData** de uma faixa aberta, null quando a faixa está fechada.
OnDetailExpandChange	Uma faixa foi aberta ou fechada.
OnDetailCellClick	Uma célula de uma faixa foi clicada: linha mestre, linha de detalhe e coluna de detalhe.
OnGetDetailData	Disparado antes de uma faixa ser preenchida. Grave as linhas em `aData` e defina `aHandled` como true para preencher a faixa por código; caso contrário a faixa é preenchida a partir de **Source**.

O estado expandido é mantido por chave mestre. Quando as linhas da origem mudam, cada faixa aberta é preenchida novamente na thread da interface, no lugar. Clicar em um cabeçalho de uma faixa a ordena, a roda do mouse sobre uma faixa a rola, e Down passa por uma faixa. A imagem da faixa é desenhada por meio de esegece.sgcGrid.Rendering, que todo pacote de grade referencia.
Quando **Source** é um componente, liberá-lo limpa **Source**. Para qualquer outro objeto, defina **Source** como null antes que o objeto desapareça.

TsgcGridDataStore

TsgcGridDataStore é um componente não visual que guarda linhas de detalhe: **Columns**, **Data** e **IsgcUITable**. Preencha-o por código, ou defina-o como o **Grid** de um **TsgcGridSource_DataSet** para carregar nele um `DataTable`. Defina **Active** como true depois de **Grid** e **DataSet**, e **LiveUpdates** mantém o armazenamento sincronizado, veja Vinculação de DataTable, JSON e CSV no .NET.

Exemplo

Um organograma como árvore, e clientes com uma faixa de seus pedidos a partir de um **TsgcGridDataStore**:
A segunda classe preenche as faixas somente por código: sem **Source** as faixas usam as **Columns** das opções, e cada linha vem de **OnGetDetailData**.

Sem um controle

TsgcGridLineView aceita as mesmas opções: atribua um **TsgcGridTreeOptions** a **TreeOptions** e um **TsgcGridDetailOptions** a **DetailOptions**; **SetNodeExpanded** e **SetDetailExpanded** alteram o estado. A exportação para Excel usa isso para gravar uma árvore sem um controle, veja Exportação para Excel do .NET.

Veja também

- Classes do .NET
- Agrupamento e Resumos do .NET
- Exportação para Excel do .NET
- Vinculação de DataTable, JSON e CSV no .NET
- Demos do .NET
- TsgcGridSource_DataSet
- Linhas de Árvore
- Mestre Detalhe
- TsgcGridDataStore

Exportação para Excel do .NET

Este recurso faz parte da edição Registered. Os pacotes Community não o contêm.
TsgcWinFormsGrid, **TsgcWpfGrid** e **TsgcAvaloniaGrid** salvam sua visão como uma pasta de trabalho do Excel (.xlsx) sem Excel instalado e sem biblioteca externa. O arquivo é gravado por **TsgcGridXLSXWriter** do pacote esegece.sgcGrid, que também funciona sem um controle.

O que é gravado

- As colunas visíveis em sua ordem, os títulos em uma linha de cabeçalho em negrito e as larguras das colunas.
- Números, datas e booleanos como células tipadas, com o formato numérico do Excel do **DisplayFormat** da coluna; o texto como strings compartilhadas.
- Os grupos como linhas de grupo com seu título e fórmulas `SUBTOTAL` para os resumos das colunas, acima de suas linhas e em níveis de estrutura de tópicos do Excel. Os grupos recolhidos são gravados como linhas ocultas, de modo que a pasta de trabalho abre como a grade a mostra.
- Quando uma coluna tem um resumo, uma linha de rodapé sob os dados com `SUBTOTAL` sobre todo o intervalo de dados.
- As linhas de árvore recuadas na coluna da árvore e em níveis de estrutura de tópicos; as linhas dos nós recolhidos são linhas ocultas.
- As colunas congeladas e a linha de cabeçalho como painel congelado.

As faixas de detalhe não são gravadas. A ordenação e o filtro se aplicam: a pasta de trabalho contém as linhas da visão.

SaveToXLSX

`SaveToXLSX(Stream aStream, string aSheetName = "Sheet1")` e `SaveToXLSX(string aFileName, string aSheetName = "Sheet1")` gravam a visão com as larguras de coluna na tela e **FrozenColumns**. Chame-os na thread da interface.

TsgcGridXLSXWriter

Membro	Descrição
SheetName	Nome da planilha, `Sheet1` por padrão. Os caracteres que o Excel rejeita viram `_` e o nome é cortado em 31 caracteres.
FrozenColumns	As primeiras colunas visíveis mantidas à vista, 0 por padrão.
Summaries	Grava as linhas `SUBTOTAL` de grupo e de rodapé, true por padrão.
FileTime	Data e hora das entradas do arquivo, a hora de criação por padrão.
SetColumnWidth	`SetColumnWidth(aCol, aDIPs)` : largura de uma coluna (índice em **Columns**) em pixels independentes de dispositivo, usada no lugar de sua **Width**. Um valor negativo restaura **Width**.
SaveToStream, SaveToFile	`SaveToStream(aData, aLines, aStream)` e `SaveToFile(aData, aLines, aFileName)` gravam a pasta de trabalho de `aData` . `aLines` fornece o agrupamento, a árvore e o estado expandido; null usa uma visão em linhas própria, construída a partir do **GroupIndex** das colunas.

Os erros levantam **EsgcGridXLSXError**. `sgcGrid_XLSX.sgcGridExcelNumberFormat` retorna o formato numérico do Excel usado para um tipo de dados e um **DisplayFormat**.

Exemplo sem um controle

Um programa de console que grava uma pasta de trabalho agrupada com o primeiro grupo recolhido, e um organograma como árvore. Ele usa o método `CreateSales` do exemplo em Agrupamento e Resumos do .NET: 100.000 linhas são gravadas em cerca de 1,4 segundo no .NET 8.

Veja também

- Agrupamento e Resumos do .NET
- Linhas de Árvore e Mestre Detalhe do .NET
- Classes do .NET
- Diferenças do .NET em Relação ao Delphi
- Demos do .NET
- Exportação para Excel

Demos do .NET

As demos estão em `.net\demos`, na solução `sgcGrid_Demos.sln`. Cada demo tem um README.md e uma opção `--test` que executa uma autoverificação fora da tela e termina com 0 em caso de sucesso, como a opção `/test` das demos do Delphi.

Demo	Host	Mostra
LiveTicker	WinForms	1000 símbolos atualizados a partir de uma thread de trabalho, com flash de alterações, cores de alta e de baixa e **TailMode**.
LiveTickerWpf	WPF	O mesmo ticker no **TsgcWpfGrid**.
LiveTickerAvalonia	Avalonia	O mesmo ticker no **TsgcAvaloniaGrid**.
DataSetGrid	WinForms	Um DataTable vinculado por meio do **TsgcGridSource_DataSet**, com importação e exportação de JSON e CSV.
Editing	WinForms	Edição no local com **OnValidate**. Somente na edição Registered.
GridAndChart	WinForms	As linhas e a seleção da grade controlam um gráfico do sgcCharts por meio do **TsgcChartDataSource_Table**, dentro de tiles do sgcDashboard.
AskYourGrid	WinForms	**TsgcGridAIQuery** com o analisador baseado em regras. Somente na edição Registered.
LiveFeed	WinForms	**TsgcGridFeedSource_WebSocket** alimentado por um **TsgcWebSocketServer** no mesmo processo. Somente na edição Registered.
Grouping	WinForms	100.000 pedidos agrupados por região e categoria, com o painel de agrupamento, resumos de grupo e de rodapé, duas colunas congeladas e atualizações de preço ao vivo a partir de uma thread de trabalho.
GroupingWpf	WPF	A mesma demo de agrupamento em **TsgcWpfGrid**.
GroupingAvalonia	Avalonia	A mesma demo de agrupamento em **TsgcAvaloniaGrid**.
TreeAndDetail	WinForms	Um organograma como linhas de árvore, e clientes de um DataTable com uma faixa de detalhe de seus pedidos a partir de um **TsgcGridDataStore**. Somente na edição Registered.
ExcelExport	Console	Grava uma pasta de trabalho de vendas agrupadas e um organograma em árvore com **TsgcGridXLSXWriter**, sem um controle. Executa no Windows e no Linux. Somente na edição Registered.

Compilando as demos

O zip das demos usa por padrão os pacotes NuGet da edição Community. Compile as demos com os pacotes Registered definindo **SgcGridEdition**:

```
dotnet build sgcGrid_Demos.sln -p:SgcGridEdition=Registered
```

Na edição Community, as demos exclusivas da Registered ainda compilam e explicam o limite quando são executadas.

Veja também

- Visão Geral do .NET
- Instalação pelo NuGet
- Início Rápido WinForms
- Feeds em Tempo Real no .NET
- Pergunte à Sua Grade no .NET
- Agrupamento e Resumos do .NET
- Linhas de Árvore e Mestre Detalhe do .NET
- Exportação para Excel do .NET
- Demos do Delphi

Solução de Problemas

Identificador não declarado StartEdit, OnValidate ou OnEdited

A edição Community é compilada sem SGC_GRID_EDIT, portanto os membros de edição das grades não existem e o código que os usa falha com E2003 Undeclared identifier. Um formulário salvo por uma compilação Registered com eventos de edição gera uma exceção Error reading ao ser carregado. Remova o código de edição, ou use uma compilação Registered ou Trial. Consulte Edições.

File not found sgcGridWS_Source ou sgcWebSocket

As fontes de feed e a consulta de IA estão na ponte sgcGridWS. A edição Community não a distribui. Nas outras edições, a ponte precisa do sgcWebSockets: adicione seus fontes ao caminho de busca, ou instale o pacote sgcWebSockets antes do pacote sgcGridWS da mesma versão. Consulte Plataformas.

A ponte não compila no Delphi 7

No Delphi 7, as units do sgcGridWS são compiladas com o sgcWebSockets, que precisa da versão do Indy para a qual foi criado. Instale a versão do Indy que a sua compilação do sgcWebSockets espera antes de compilar a ponte.

O demo DataSetGrid do C++Builder falha no Win64

O Win32 vincula o MIDAS estaticamente, mas o C++Builder não tem biblioteca MIDAS estática para Win64. Copie midas.dll para a mesma pasta do executável de 64 bits.

Os valores não mudam ou as linhas são duplicadas

UpdateRow localiza as linhas pela chave, e as chaves diferenciam maiúsculas de minúsculas. Use a mesma chave em AppendRow e UpdateRow. Com AutoInsert False, uma atualização de uma chave ausente é ignorada. Nomes de campos que não correspondem a uma coluna também são ignorados.

OnFilterRow lê a linha errada

OnFilterRow recebe uma linha do armazenamento. Leia-a com StoreCells ou StoreKeyOf, e não com Cells, que recebe linhas da exibição. Consulte Filtragem.

A grade trava quando um manipulador de feed mexe em um controle

OnFilterRow e OnGetValue são executados com o bloqueio dos dados mantido, e OnChange, OnRow e OnError dos feeds são executados na thread de escrita. Nunca espere pela thread principal ali e nunca mexa em um controle; use TThread.Queue.

Números mostram vírgula ou ponto de forma inesperada

Os valores são formatados com Data.FormatSettings, que segue a localidade. Chame Data.UseInvariantFormat para um separador decimal ponto, ou defina FormatSettings.

Identificador não declarado Tree, Detail ou SaveToXLSX

As linhas de árvore, as linhas mestre-detalhe, o TsgcGridDataStore e a exportação para Excel são compilados apenas com SGC_GRID_ADVANCED, que a edição Community deixa de fora. O código que os usa falha com E2003 Undeclared identifier, e um formulário salvo com propriedades Tree ou Detail dispara uma exceção Error reading ao ser carregado. Veja Edições.

A árvore mostra uma lista plana

Tree.KeyField e Tree.ParentField devem indicar ambos colunas da grade (elas podem estar ocultas), e os dados não podem ser virtuais. Os valores de pai são comparados com os textos de chave diferenciando maiúsculas de minúsculas; uma linha cujo pai não é encontrado torna-se uma raiz. Veja Linhas de Árvore.

O agrupamento não tem efeito

O agrupamento é ignorado enquanto o modo de árvore está ativo. Limpe Tree.KeyField ou Tree.ParentField para agrupar de novo.

Nenhum expansor de detalhe é mostrado

Uma linha só recebe um expansor quando Detail.Active é True, Detail.MasterKeyField indica uma coluna da grade, um Detail.Source ou um manipulador OnGetDetailData está definido, e a chave mestre da linha não é vazia. Dados virtuais não mostram faixas. Veja Mestre Detalhe.

Uma faixa de detalhe permanece vazia

Detail.DetailKeyField deve indicar uma coluna da origem, e seus valores devem coincidir com os textos da chave mestre (diferenciando maiúsculas de minúsculas; 5, 5.0 e '5' são a mesma chave). Uma origem virtual não fornece linhas. Com OnGetDetailData e sem Source, defina Detail.Columns, porque a faixa obtém suas colunas dali.

As linhas de detalhe não aparecem no arquivo Excel

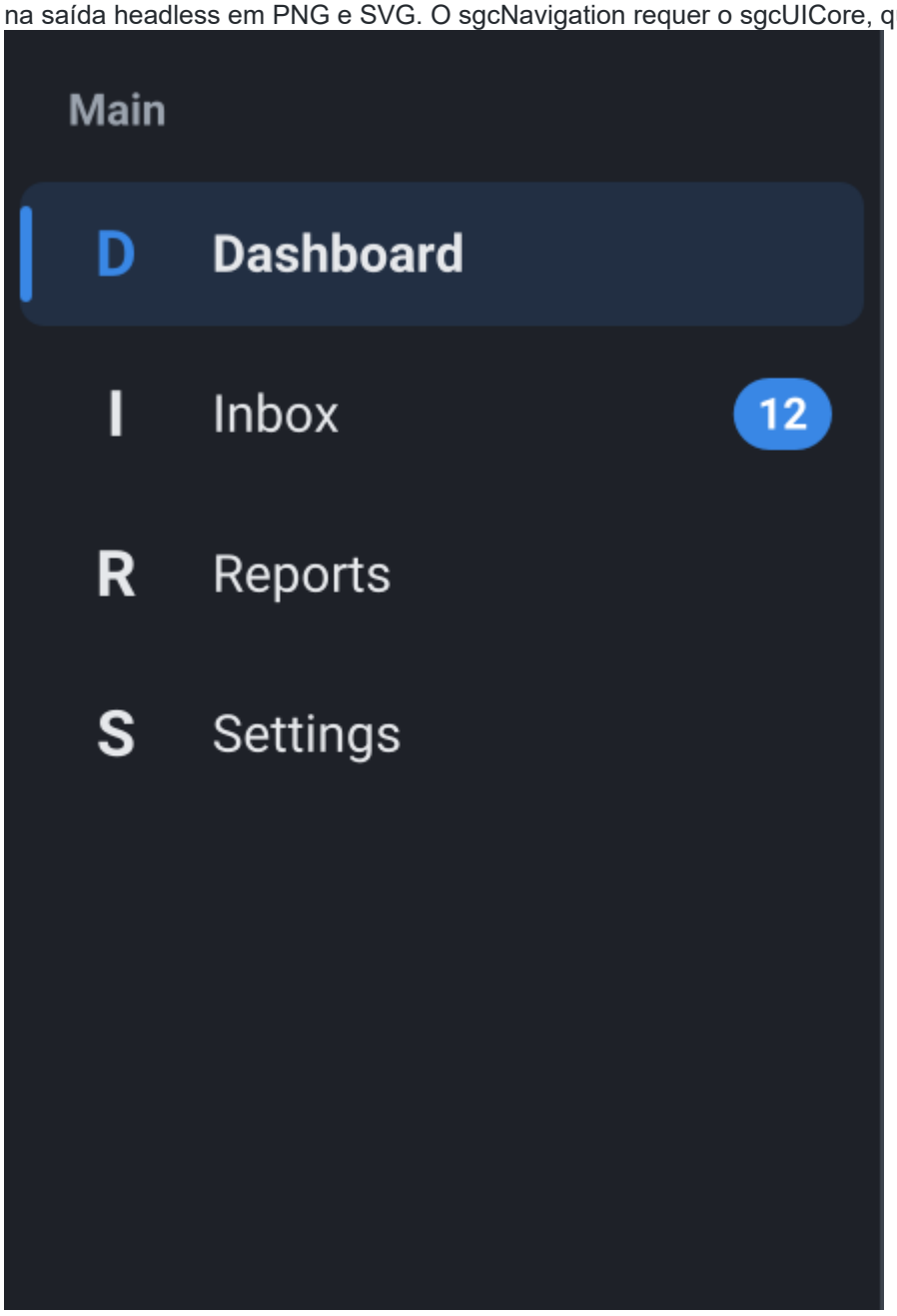
SaveToXLSX não grava faixas de detalhe. Exporte a origem de detalhe separadamente, por exemplo com um TsgcGridXLSXWriter sobre o Data do TsgcGridDataStore. Veja Exportação para Excel.

SaveToXLSX dispara EsgcGridXLSXError

Uma planilha XLSX contém no máximo 1,048,576 linhas e 16,384 colunas, linhas de grupo e cabeçalho incluídos. Filtre as linhas ou oculte colunas antes da exportação. Nada é gravado quando a exceção é disparada.

Introdução

sgcNavigation é uma barra de navegação nativa para Delphi e C++Builder, para VCL e FireMonkey. Um único componente, **TsgcVCLNavigationBar** ou **TsgcFMXNavigationBar**, exibe as páginas de um aplicativo como uma lista vertical de itens com ícones, legendas, selos e cabeçalhos de grupo. No modo sidebar, é uma lista plana que se recolhe em um trilho de ícones com animação. No modo accordion, os cabeçalhos de grupo expandem e recolhem os itens abaixo deles.
O sgcNavigation faz parte da suíte visual da eSeGeCe e emoldura os outros produtos: uma barra de navegação à esquerda e, à direita, um dashboard **sgcDashboard**, uma grade **sgcGrid** ou uma página com um **sgcToolbar**. Ele é desenhado com o **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro, de modo que a barra tem a mesma aparência em VCL, em FireMonkey e na saída headless em PNG e SVG. O sgcNavigation requer o sgcUICore, que é instalado junto com ele.



Edições

O sgcNavigation é distribuído nas edições Registered, Trial e Community. A edição Community tem todos os recursos, exceto os Action links dos itens. Consulte Edições.

Por onde começar

- Arquitetura: o modelo independente de framework, o renderizador sobre o canvas do sgcUICore e os controles VCL e FMX.
- Plataformas: as versões do Delphi, o C++Builder e as plataformas.
- Início Rápido VCL, Início Rápido FMX e Início Rápido C++Builder: uma primeira barra de navegação em poucos minutos.
- Componentes: as barras de navegação, os itens e o modelo.
- Modo sidebar, Modo accordion, Recolhimento e animação, Imagens e glifos, Selos, Teclado e foco, Action links, Temas e Renderização headless.
- Demos: os projetos de demonstração em Delphi e C++Builder.

Arquitetura

O sgcNavigation é construído em camadas. Cada camada depende apenas das que estão abaixo dela, de modo que o modelo e o renderizador compilam sem VCL nem FireMonkey.

Camada	Unit	Função
Modelo	sgcNavigation_Model	TsgcNavigationModel guarda os Items (uma coleção de TsgcNavigationItem), o modo, o estado recolhido, os tamanhos, o tema e o estado de interação (item selecionado, sob o mouse, pressionado e com foco). Seu método Layout calcula a linha de cada item exibido e HitTest encontra o item sob um ponto.
Renderizador	sgcNavigation_Render	sgcNavigationRender desenha um modelo em qualquer canvas do sgcUICore. sgcNavigationSaveToPNG e sgcNavigationSaveToSVG o usam para gravar uma imagem sem janela. Consulte Renderização headless.
Controle VCL	sgcNavigation_VCL	TsgcVCLNavigationBar possui um modelo, publica suas propriedades e o desenha por meio da base de controles VCL do sgcUICore. Ele acrescenta o mouse, o teclado, as dicas, a lista de imagens e a animação de recolhimento.
Controle FMX	sgcNavigation_FMX	TsgcFMXNavigationBar é o controle FireMonkey com as mesmas propriedades, métodos e eventos, além de entrada por toque.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme	O runtime compartilhado: o canvas vetorial e o rasterizador, a base de controles VCL e FMX com ritmo de quadros (MaxFPS) e os tokens dos temas claro e escuro.

Modelo e controle

As propriedades publicadas de uma barra de navegação são as propriedades do seu modelo, e a propriedade **Model** da barra dá acesso ao restante: ToggleGroup, HitTest, NextVisible e o estado de interação. Uma alteração no modelo agenda um novo quadro e nunca desenha diretamente, de modo que várias alterações seguidas custam uma única repintura.

Desenho

Cada linha é desenhada pelo renderizador: o espaço do ícone com uma imagem ou um glifo, a legenda ajustada à largura livre, o selo e os estados sob o mouse, pressionado, selecionado, com foco e desabilitado. Todos os tamanhos são pixels lógicos multiplicados pela escala do monitor, de modo que a barra fica nítida em qualquer DPI. A barra VCL usa PPI / 96 como escala e a barra FMX usa a escala do formulário.

Veja Também

- Plataformas
- TsgcNavigationModel
- Temas

Edições

O sgcNavigation é compilado a partir do mesmo código-fonte para todas as edições. A edição só altera quais recursos uma compilação inclui.

Edição	Inclui
Registered	Tudo: as barras de navegação VCL e FMX, os modos sidebar e accordion, o trilho de ícones recolhido com sua animação, imagens, glifos e selos, navegação pelo teclado, temas, os Action links dos itens e a saída headless em PNG e SVG.
Trial	Os mesmos recursos da Registered. Uma compilação feita com SGC_TRIAL exibe um aviso de que é uma versão de avaliação depois que a janela de lançamento indicada na compilação tiver passado. Nada mais muda.
Community	Tudo, exceto os Action links dos itens.

O que a Community deixa de fora

A edição Community compila com SGC_NAVIGATION_ACTIONS indefinido. Em uma compilação Community, a propriedade **Action** de TsgcNavigationItem não existe, portanto um item não pode ser vinculado a uma TAction. Atribua o **OnClick** do item, ou o evento **OnItemClick** da barra.
A edição é uma escolha em tempo de compilação no sgcVer.inc (SGC_COMMUNITY, que também desativa SGC_TRIAL), não uma chave em tempo de execução. Código ou um arquivo de formulário que use Action não compila nem carrega em uma compilação Community, consulte Solução de Problemas.

Veja Também

- Action links
- Plataformas

Instalação

O sgcNavigation faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcNavigation** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcNavigation

sgcUICore, a base compartilhada dos produtos de layout e do sgcCharts, acompanha o sgcNavigation. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcNavigation é selecionado. O sgcNavigation não tem pacote de ponte sgcWebSockets.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13` .
`Demos\Navigation`	Os demos do sgcNavigation. Veja Demos.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcNavigation.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcNavigation. Os seus produtos são sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard, enquanto o sgcPDF e o sgcMaps não fazem parte do Community. Veja Edições para saber o que a compilação Community do sgcNavigation deixa de fora.

Instalações autônomas anteriores

Se o instalador encontrar uma instalação autônoma mais antiga do sgcUICore, como a gravada pelo instalador autônomo do sgcCharts, ele lista os library paths e os pacotes de design dessa instalação na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcNavigationD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcNavigation<V>` / `dclsgcNavigation<V>`	Os componentes do sgcNavigation (runtime / design time).

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcNavigation`, depois `dclsgcNavigation`. Em seguida, instale o pacote de design time `dclsgcNavigation` e adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcNavigationC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

.NET

Nada muda para o .NET. O sgcNavigation continua sendo distribuído como seus próprios pacotes NuGet (`esegece.sgcNavigation` e os pacotes complementares). Veja a instalação do .NET.

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	TsgcVCLNavigationBar no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	TsgcFMXNavigationBar a partir do Delphi XE6 (SGC_NAVIGATION_FMX), em Win32, Win64, macOS, iOS e Android, e no Linux com FMX for Linux (FMXLinux). A propriedade Images requer o XE8 ou posterior (FMX.ImgList), e as dicas dos itens requerem o 10.1 Berlin ou posterior.
Linux64	As units independentes de framework, o modelo e o renderizador com a saída headless em PNG e SVG, a partir do Delphi 10.2 (SGC_NAVIGATION_LINUX).
.NET	Um port em C# com a mesma API para WinForms, WPF e Avalonia, e a saída headless em PNG e SVG, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão Geral do .NET.

Pacotes

O sgcNavigation traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcNavigationD13.dpk para o Delphi 13 e sgcNavigationC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcNavigation	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, sgcUICore
sgcNavigationVCL	Runtime, VCL	sgcNavigation_VCL (TsgcVCLNavigationBar)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcNavigation
sgcNavigationFMX	Runtime, FireMonkey, a partir do XE6	sgcNavigation_FMX (TsgcFMXNavigationBar)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcNavigation
dclsgcNavigation	Tempo de projeto, VCL	Registra TsgcVCLNavigationBar na página **SGC Navigation** da paleta.	designide, rtl, vcl, sgcNavigation, sgcNavigationVCL
dclsgcNavigationFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXNavigationBar na página **SGC Navigation** da paleta.	designide, rtl, fmx, sgcNavigation, sgcNavigationFMX, dclsgcNavigation

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcNavigationFMX exige o dclsgcNavigation, portanto instale primeiro o dclsgcNavigation.

Versões do IDE e plataformas de destino

Delphi	sgcNavigation	sgcNavigationVCL	sgcNavigationFMX	dclsgcNavigation, dclsgcNavigationFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcNavigation, sgcNavigationVCL	sgcNavigationFMX	dclsgcNavigation, dclsgcNavigationFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcNavigation)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcNavigation)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcNavigationC13.cbproj, sgcNavigationVCLC13.cbproj, sgcNavigationFMXC13.cbproj, dclsgcNavigationC13.cbproj e dclsgcNavigationFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXNavigationBar) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcNavigationVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcNavigation_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcNavigation é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa as unidades independentes de framework. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.
A barra VCL só aparece na paleta de um formulário VCL e a barra FMX só na paleta de um formulário FireMonkey.
A propriedade **Items** é editada com o editor de coleções padrão.

Veja Também

- Edições
- Arquitetura
- Solução de Problemas

Início Rápido VCL

Em tempo de design

- Solte um **TsgcVCLNavigationBar** da página **SGC Navigation** da paleta em um formulário. Ele é alinhado à esquerda por padrão.
- Dê um duplo clique em **Items** no Object Inspector e adicione os itens. Defina **Caption** e **Glyph** (uma letra ou um símbolo curto) para cada item, e defina **IsGroup** como True em um item que seja um cabeçalho de grupo.
- Solte um TPanel ou um TPageControl no restante do formulário com **Align** definido como alClient.
- Escreva um manipulador de **OnItemClick** que mostre a página do item clicado, ou atribua o **OnClick** de cada item.
- Defina **Theme** como utDark para a paleta escura e **Collapsed** como True para ver o trilho de ícones.

Em código

A mesma barra criada em tempo de execução. O programa compila como está com `sgcUI\Source` no caminho de busca.

```
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;
```

Veja Também

- TsgcVCLNavigationBar
- TsgcNavigationItem
- Modo sidebar
- Recolhimento e animação
- Temas

Início Rápido FMX

O controle FireMonkey requer o Delphi XE6 ou posterior. Ele é usado exatamente como o da VCL: solte um **TsgcFMXNavigationBar**, que é alinhado à esquerda por padrão, preencha **Items** com o editor de coleções e trate **OnItemClick**. Um toque em uma linha funciona como um clique.

Em 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;
```

Veja Também

- TsgcFMXNavigationBar
- TsgcNavigationItem
- Plataformas

Início Rápido C++Builder

Instale os pacotes do sgcNavigation para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página **SGC Navigation** da paleta exatamente como no Delphi. O instalador adiciona os dois caminhos à IDE para você. Os cabeçalhos .hpp e os arquivos .lib gerados do sgcNavigation e do sgcUICore estão nessa única pasta, portanto o sgcUICore não precisa de um caminho próprio.

Em código

OnItemClick é um evento de closure, portanto seu manipulador é uma função membro __fastcall de uma classe.

Veja Também

- Início Rápido VCL
- Demos
- Plataformas

Demos

As demos estão na pasta **Demos** da instalação, com os projetos Delphi em Demos\Delphi e as versões C++Builder das demos VCL em Demos\CBuilder. Cada pasta de demo tem um readme.html que explica o que a demo faz e como funciona.

Demo	Componentes	O que mostra
SidebarApp	TsgcVCLNavigationBar, TsgcVCLDashboard, TsgcVCLKPICard	Uma barra de navegação no modo sidebar com ícones, selos, um item desabilitado e um recolhimento animado, controlando as páginas de um TPageControl, uma delas um dashboard sgcDashboard. Delphi e C++Builder, Win32 e Win64. Requer o sgcDashboard.
Accordion	TsgcVCLNavigationBar, TActionList	Uma barra de navegação no modo accordion com três grupos, uma chave de expansão única ou múltipla e itens vinculados às actions de uma TActionList. Delphi e C++Builder, Win32 e Win64. Os Action links não fazem parte da edição Community, portanto esta demo não compila com ela.
SidebarFMX	TsgcFMXNavigationBar, TsgcFMXDashboard, TsgcFMXKPICard	A versão FireMonkey da SidebarApp: glifos, selos, um item desabilitado e um recolhimento animado, controlando as abas de um TTabControl, uma delas um dashboard sgcDashboard. Delphi, Win32 e Win64. Requer o sgcDashboard.

Veja Também

- Início Rápido VCL
- Modo accordion
- Action links

TsgcVCLNavigationBar

Unit: sgcNavigation_VCL
TsgcVCLNavigationBar é a barra de navegação VCL, somente para Windows. Ela é alinhada à esquerda por padrão e exibe seus Items como linhas, no modo sidebar ou no modo accordion. Ela trata o mouse, a roda, o teclado e as dicas, e rola quando as linhas são mais altas que o controle. Todos os tamanhos são pixels lógicos multiplicados por PPI / 96.

Propriedades

Propriedade	Tipo	Descrição
Items	TsgcNavigationItems	Os itens e cabeçalhos de grupo, na ordem de exibição. Consulte TsgcNavigationItems.
Mode	TsgcNavigationMode	nbmSidebar (padrão), uma lista plana, ou nbmAccordion, grupos que expandem e recolhem.
Collapsed	Boolean	Mostra a barra como um trilho de ícones com CollapsedWidth de largura em vez de ExpandedWidth. O padrão é False.
ExpandedWidth	Integer	Largura da barra expandida em pixels lógicos. O padrão é 220.
CollapsedWidth	Integer	Largura do trilho de ícones em pixels lógicos. O padrão é 56.
ItemHeight	Integer	Altura de uma linha de item em pixels lógicos. O padrão é 40.
GroupHeight	Integer	Altura de uma linha de cabeçalho de grupo em pixels lógicos. O padrão é 32.
Indent	Integer	Recuo dos itens de um grupo no modo accordion, em pixels lógicos. O padrão é 16.
AllowMultiExpand	Boolean	No modo accordion, permite que mais de um grupo fique expandido. Quando False (padrão), expandir um grupo recolhe os outros.
SelectedIndex	Integer	Índice do item selecionado, -1 (padrão) quando não há nenhum. Um índice fora do intervalo vira -1.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom, que usa as cores do tema claro.
AccentColor	TsgcUIColor	Cor da seleção, do foco e dos selos no formato $AARRGGBB. 0 (padrão) usa o destaque do tema.
Images	TCustomImageList	A lista de imagens do ImageIndex dos itens. As imagens são desenhadas no seu próprio tamanho.
AnimateCollapse	Boolean	Anima a largura por cerca de 150 ms quando Collapsed muda em tempo de execução. O padrão é True.
Model	TsgcNavigationModel (public, somente leitura)	O modelo da barra. Consulte TsgcNavigationModel.

ScrollY	Integer (public, somente leitura)	O deslocamento de rolagem vertical em pixels do dispositivo.

Os membros habituais de um controle também são publicados: **Align** (padrão alLeft), **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop** (padrão True), **Visible**, **OnClick**, **OnContextPopup**, **OnEnter**, **OnExit**, **OnKeyDown**, **OnKeyUp**, **OnMouseDown**, **OnMouseMove**, **OnMouseUp** e **OnResize**.

Métodos

Método	Retorna	Descrição
ItemAtPos(X, Y: Integer)	Integer	O item em um ponto do controle, -1 quando não há nenhum. Uma sobrecarga também retorna a parte sob o ponto, npItem ou npExpander.
ItemRect(aIndex: Integer)	TRect	A linha de um item em coordenadas do controle, com a rolagem aplicada, ou um retângulo vazio quando o item não é exibido.
ScrollIntoView(aIndex: Integer)		Rola a barra para que a linha do item fique visível.
InitiateAction		Permite que cada item com uma Action se atualize. O framework o chama quando está ocioso.

Eventos

Evento	Tipo	Descrição
OnItemClick	TsgcNavigationItemEvent	Um item habilitado foi clicado, tocado ou ativado com Enter ou Espaço. O item já está selecionado. O OnClick do item ou sua Action é executado depois deste evento.
OnSelectionChange	TNotifyEvent	SelectedIndex mudou, pelo usuário ou por código.
OnCollapsedChange	TNotifyEvent	Collapsed mudou.

Exemplo

```
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;
```

Veja Também

- TsgcFMXNavigationBar
- TsgcNavigationItem
- TsgcNavigationModel
- Teclado e foco

TsgcFMXNavigationBar

Unit: sgcNavigation_FMX
TsgcFMXNavigationBar é a barra de navegação FireMonkey, a partir do Delphi XE6, para Windows, macOS, iOS, Android e Linux com FMXLinux. Ela tem as propriedades, métodos e eventos de TsgcVCLNavigationBar e acrescenta entrada por toque: um toque em uma linha funciona como um clique. Os tamanhos são unidades lógicas multiplicadas pela escala do formulário. A partir do 10.1 Berlin, o Hint do controle acompanha o item sob o cursor.

Propriedades

Propriedade	Tipo	Descrição
Items	TsgcNavigationItems	Os itens e cabeçalhos de grupo, na ordem de exibição. Consulte TsgcNavigationItems.
Mode	TsgcNavigationMode	nbmSidebar (padrão), uma lista plana, ou nbmAccordion, grupos que expandem e recolhem.
Collapsed	Boolean	Mostra a barra como um trilho de ícones com CollapsedWidth de largura em vez de ExpandedWidth. O padrão é False.
ExpandedWidth	Integer	Largura da barra expandida em pixels lógicos. O padrão é 220.
CollapsedWidth	Integer	Largura do trilho de ícones em pixels lógicos. O padrão é 56.
ItemHeight	Integer	Altura de uma linha de item em pixels lógicos. O padrão é 40.
GroupHeight	Integer	Altura de uma linha de cabeçalho de grupo em pixels lógicos. O padrão é 32.
Indent	Integer	Recuo dos itens de um grupo no modo accordion, em pixels lógicos. O padrão é 16.
AllowMultiExpand	Boolean	No modo accordion, permite que mais de um grupo fique expandido. Quando False (padrão), expandir um grupo recolhe os outros.
SelectedIndex	Integer	Índice do item selecionado, -1 (padrão) quando não há nenhum. Um índice fora do intervalo vira -1.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom, que usa as cores do tema claro.
AccentColor	TsgcUIColor	Cor da seleção, do foco e dos selos no formato $AARRGGBB. 0 (padrão) usa o destaque do tema.
Images	TCustomImageList	A lista de imagens do ImageIndex dos itens. No FMX ela requer o XE8 ou posterior e as imagens são desenhadas com 16 x 16 pixels lógicos.
AnimateCollapse	Boolean	Anima a largura por cerca de 150 ms quando Collapsed muda em tempo de execução. O padrão é True.
Model	TsgcNavigationModel (public, somente leitura)	O modelo da barra. Consulte TsgcNavigationModel.

ScrollY	Integer (public, somente leitura)	O deslocamento de rolagem vertical em pixels do dispositivo.

Os membros habituais de um controle FMX também são publicados, entre eles **Align** (padrão Left), **Anchors**, **CanFocus** (padrão True), **Enabled**, **HitTest**, **Margins**, **Padding**, **Opacity**, **PopupMenu**, **Position**, **Width**, **Height**, **TabOrder**, **Visible**, **ShowHint** e **TabStop** (10.1 e posteriores), e os eventos OnClick, OnEnter, OnExit, OnKey, OnMouse, OnPaint e OnResize.

Métodos

Método	Retorna	Descrição
ItemAtPos(X, Y: Single)	Integer	O item em um ponto do controle, -1 quando não há nenhum. Uma sobrecarga também retorna a parte sob o ponto, npItem ou npExpander.
ItemRect(aIndex: Integer)	TRectF	A linha de um item em coordenadas do controle, com a rolagem aplicada, ou um retângulo vazio quando o item não é exibido.
ScrollIntoView(aIndex: Integer)		Rola a barra para que a linha do item fique visível.
InitiateAction		Permite que cada item com uma Action se atualize. O framework o chama quando está ocioso.

Eventos

Evento	Tipo	Descrição
OnItemClick	TsgcNavigationItemEvent	Um item habilitado foi clicado, tocado ou ativado com Enter ou Espaço. O item já está selecionado. O OnClick do item ou sua Action é executado depois deste evento.
OnSelectionChange	TNotifyEvent	SelectedIndex mudou, pelo usuário ou por código.
OnCollapsedChange	TNotifyEvent	Collapsed mudou.

Exemplo

```
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;
```

Veja Também

- TsgcVCLNavigationBar
- TsgcNavigationItem
- Plataformas

TsgcNavigationItem

Unit: sgcNavigation_Model
TsgcNavigationItem é uma linha de uma barra de navegação: uma página do aplicativo ou um cabeçalho de grupo quando IsGroup é True. Os itens ficam na coleção Items da barra.

Propriedades

Propriedade	Tipo	Descrição
Caption	string	O texto da linha. Fica oculto no trilho de ícones recolhido.
Glyph	string	Um texto curto ou símbolo, por exemplo uma letra, desenhado centralizado no espaço do ícone quando o item não tem imagem.
ImageIndex	Integer	Índice da imagem em Images da barra. O padrão é -1, sem imagem.
Badge	string	Um texto curto desenhado como uma pílula de destaque à direita da linha, e como um ponto sobre o ícone no trilho. Vazio (padrão) não desenha selo. Consulte Selos.
IsGroup	Boolean	Transforma o item em um cabeçalho de grupo. Um cabeçalho nunca é selecionado. No modo accordion, ele expande e recolhe os itens abaixo dele até o próximo cabeçalho. O padrão é False.
Expanded	Boolean	Define se os itens de um cabeçalho de grupo são exibidos no modo accordion. O padrão é True.
Enabled	Boolean	Um item desabilitado é desenhado esmaecido e não pode ser selecionado nem clicado. O padrão é True.
Visible	Boolean	Um item oculto não recebe linha. O padrão é True.
Hint	string	A dica da linha. No trilho recolhido, um item sem Hint mostra sua Caption como dica.
Tag	NativeInt	Um valor livre para o aplicativo, Integer antes do XE2.
Action	TBasicAction	Uma action vinculada. Caption, Enabled, Visible, Hint e ImageIndex a acompanham e um clique a executa, a menos que OnClick esteja atribuído. Não existe na edição Community. Consulte Action links.
BoundsRect	TsgcUIRect (public, somente leitura)	A linha do último layout em pixels do dispositivo, em coordenadas de conteúdo antes da rolagem.
Shown	Boolean (public, somente leitura)	Indica se o último layout deu uma linha ao item.
Separator	Boolean (public,	Indica se a linha é uma linha separadora de um grupo no trilho.

	somente leitura)	
Level	Integer (public, somente leitura)	O nível de recuo da linha, 1 para os itens de um grupo accordion.

Métodos

Método	Retorna	Descrição
Click(aComponent: TComponent = nil)		Executa o item: seu OnClick quando atribuído, senão sua Action, executada com aComponent como ActionComponent. A barra o chama depois de OnItemClick.
InitiateAction		Permite que a Action vinculada se atualize. Não faz nada sem uma Action.

Eventos

Evento	Tipo	Descrição
OnClick	TNotifyEvent	O item foi ativado. Sender é o item.

Exemplo

```
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;
```

Veja Também

- TsgcNavigationItems
- Imagens e glifos
- Selos
- Action links

TsgcNavigationItems

Unit: sgcNavigation_Model
TsgcNavigationItems é a coleção de TsgcNavigationItem de uma barra de navegação, uma TOwnedCollection, de modo que os itens são editados em tempo de design com o editor de coleções padrão e ficam armazenados no arquivo de formulário. A ordem da coleção é a ordem das linhas. Um cabeçalho de grupo possui os itens que o seguem até o próximo cabeçalho.

Métodos

Método	Retorna	Descrição
Add	TsgcNavigationItem	Adiciona um item vazio no final.
AddItem(const aCaption: string; const aGlyph: string = ''; aIsGroup: Boolean = False)	TsgcNavigationItem	Adiciona um item, ou um cabeçalho de grupo quando aIsGroup é True, com sua Caption e seu Glyph, e o retorna.
Items[Index: Integer]	TsgcNavigationItem	O item em um índice. É a propriedade padrão, portanto Nav.Items[2] é o terceiro item.

Os membros herdados de TCollection também estão disponíveis: **Count**, **Clear**, **Delete**, **Insert**, **BeginUpdate** e **EndUpdate**. Envolva muitas alterações em BeginUpdate e EndUpdate para calcular o layout da barra uma única vez.

Exemplo

```
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;
```

Veja Também

- TsgcNavigationItem
- Modo accordion

TsgcNavigationModel

Unit: sgcNavigation_Model
TsgcNavigationModel é o modelo independente de framework de uma barra de navegação. Toda barra possui um, publicado por meio de sua propriedade **Model**, e um modelo também pode ser criado isoladamente, sem controle, para renderizar uma barra como imagem ou para testar o layout. Ele não tem dependência de VCL nem de FMX e compila em todas as plataformas. Foi feito para ser usado a partir da thread principal.

Propriedades

As propriedades publicadas são as da barra: Items, Mode, Collapsed, ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight, Indent, AllowMultiExpand, SelectedIndex, Theme e AccentColor, consulte TsgcVCLNavigationBar. O modelo acrescenta estas propriedades públicas:

Propriedade	Tipo	Descrição
HotIndex	Integer	O item sob o mouse, -1 quando não há nenhum.
PressedIndex	Integer	O item que está sendo pressionado, -1 quando não há nenhum.
FocusedIndex	Integer	O item com o foco do teclado, -1 quando não há nenhum.
ShowFocus	Boolean	Desenha o retângulo de foco. O controle o define enquanto o teclado é usado.
LayoutWidth	Integer (somente leitura)	A largura do último Layout, em pixels do dispositivo.
LayoutScale	Double (somente leitura)	A escala do último Layout.

Métodos

Método	Retorna	Descrição
Layout(aWidth: Integer; aScale: Double = 1)		Calcula as linhas para uma barra com aWidth pixels de largura no dispositivo. aScale multiplica cada tamanho lógico.
ContentHeight	Double	Altura das linhas mais o preenchimento do último Layout, em pixels do dispositivo.
HitTest(aX, aY: Double; out aPart: TsgcNavigationPart)	Integer	O item em um ponto, em coordenadas de conteúdo do último Layout, -1 quando não há nenhum. aPart é npItem, ou npExpander no expansor de um grupo accordion. Uma sobrecarga não tem aPart. Itens desabilitados são reportados, um separador do trilho não.
IsItemShown(aIndex: Integer)	Boolean	Indica se o item recebe uma linha com o Mode, o Collapsed e o estado de grupo atuais.
IsItemFocusable(aIndex: Integer)	Boolean	Indica se o teclado pode parar no item: exibido, habilitado e que não seja um cabeçalho de grupo, exceto no accordion expandido.

NextVisible(aIndex: Integer)	Integer	O próximo item que pode receber foco, -1 quando não há nenhum. NextVisible(-1) é o primeiro.
PrevVisible(aIndex: Integer)	Integer	O item anterior que pode receber foco, -1 quando não há nenhum. PrevVisible(Items.Count) é o último.
GroupOf(aIndex: Integer)	Integer	O cabeçalho de grupo ao qual um item pertence, -1 quando o item é um grupo ou nenhum grupo o precede.
ToggleGroup(aIndex: Integer)		Expande ou recolhe um cabeçalho de grupo. Expandir recolhe os outros grupos quando AllowMultiExpand é False.
CurrentWidth	Integer	A largura lógica para o estado Collapsed atual, ExpandedWidth ou CollapsedWidth.
GetAccent	TsgcUIColor	A cor de destaque em uso: AccentColor, ou o destaque do tema quando for 0.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O modelo mudou e precisa de repintura. A barra o atribui, portanto use-o apenas em um modelo criado isoladamente.

Exemplo

```
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;
```

Veja Também

- Renderização headless
- Arquitetura
- TsgcVCLNavigationBar

Modo Sidebar

O modo sidebar, **Mode** = nbmSidebar, é o padrão. A barra mostra cada item visível como uma linha, empilhadas a partir do topo. Um cabeçalho de grupo é um rótulo estático em cor esmaecida que dá título aos itens abaixo dele, nunca é selecionado e um clique nele não faz nada.

- Uma linha de item tem ItemHeight pixels de altura e uma linha de cabeçalho de grupo tem GroupHeight pixels de altura.
- Um clique ou um toque em um item habilitado o seleciona, dispara OnItemClick e depois executa o OnClick do item ou sua Action.
- A linha selecionada recebe um fundo com tom do destaque e uma barra de destaque de 3 pixels na borda esquerda.
- Quando as linhas são mais altas que o controle, a roda do mouse rola a barra, e ScrollIntoView mostra um item específico.
- Collapsed transforma a sidebar em um trilho de ícones, consulte Recolhimento e animação.

Exemplo

```
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;
```

Veja Também

- Modo accordion
- Recolhimento e animação
- TsgcVCLNavigationBar

Modo Accordion

No modo accordion, **Mode** = nbmAccordion, um cabeçalho de grupo expande e recolhe os itens que o seguem até o próximo cabeçalho. Os itens de um grupo só são exibidos enquanto o grupo está visível e sua propriedade **Expanded** é True, e são recuados por **Indent**. O cabeçalho tem um expansor em forma de divisa (chevron) na borda direita.

- Um clique na linha do cabeçalho ou no seu expansor alterna o grupo. O teclado faz o mesmo com Enter ou Espaço, e Esquerda recolhe e Direita expande o grupo com foco.
- Com **AllowMultiExpand** False (padrão), expandir um grupo recolhe os outros, de modo que apenas um grupo fica aberto por vez. Defina como True para deixar vários grupos abertos.
- Em código, **Model.ToggleGroup** alterna um grupo com a mesma regra, e definir **Expanded** de um cabeçalho o abre ou fecha sem afetar os outros.
- Os itens antes do primeiro cabeçalho de grupo não pertencem a nenhum grupo e são sempre exibidos.
- No trilho recolhido, os grupos não são alternados: todo item visível que não seja um cabeçalho é exibido, e cada grupo vira uma linha separadora.

Exemplo

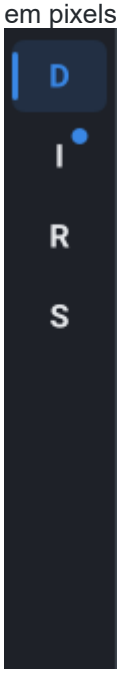
```
uses
  Forms, Classes, Controls, StdCtrls, ActnList, ImgList, Graphics,
  sgcUICore_Theme, sgcNavigation_Model, sgcNavigation_VCL;

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;
```

Veja Também

- Modo sidebar
- TsgcNavigationItems
- Demos

Recolhimento e Animação

Collapsed alterna a largura da barra entre **ExpandedWidth** (padrão 220) e **CollapsedWidth** (padrão 56), ambos em pixels lógicos. A barra recolhida é um trilho de ícones, tanto no modo sidebar quanto no modo accordion:



- Apenas o espaço do ícone de cada item é desenhado. As legendas ficam ocultas, e a dica de um item sem Hint é a sua Caption.
- Um cabeçalho de grupo vira uma linha separadora, exceto quando seria a primeira linha ou viria logo após outro separador.
- Um selo vira um ponto sobre o ícone.
- Com **AnimateCollapse** True (padrão), a largura é animada por cerca de 150 ms em tempo de execução. Em tempo de design, ou com AnimateCollapse False, ela muda de uma vez.
- **OnCollapsedChange** é disparado quando Collapsed muda.

A barra define sua própria Width, portanto deixe Align em alLeft (VCL) ou Left (FMX) e faça o conteúdo à sua direita usar alClient. O modelo informa a largura com **CurrentWidth**.

Exemplo

```
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;
```

Veja Também

- Modo sidebar
- Selos
- TsgcVCLNavigationBar

Imagens e Glifos

Cada linha tem um espaço de ícone à esquerda. Ele mostra, nesta ordem de preferência:

- a imagem **ImageIndex** da lista de imagens **Images** da barra, quando ambos estão definidos;
- senão, o texto **Glyph** do item, centralizado, por exemplo uma letra ou um símbolo curto;
- senão, nada.

Um glifo não precisa de lista de imagens e segue as cores do tema, e é por isso que as demos usam letras. Um item desabilitado desenha sua imagem com 40% de opacidade.

VCL

Images é uma TCustomImageList, e as imagens são desenhadas no seu próprio tamanho. Listas de imagens com máscara e de 32 bits funcionam: cada imagem é convertida uma vez para ARGB e mantida em cache até que a lista de imagens mude.

FireMonkey

Images é a TCustomImageList do FMX, de FMX.ImgList, e requer o Delphi XE8 ou posterior. As imagens são desenhadas com 16 x 16 pixels lógicos, obtidas da lista de imagens no tamanho de dispositivo do quadro, e mantidas em cache até que a lista de imagens mude.

Exemplo

```
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;
```

Veja Também

- TsgcNavigationItem

- Plataformas

Selos

Badge é um texto curto de um item, por exemplo uma contagem de não lidos ou a palavra New. Ele é desenhado como uma pílula na cor de destaque, alinhada à direita na linha, com texto preto ou branco, o que tiver melhor leitura sobre o destaque. Um Badge vazio não desenha nada. No trilho de ícones recolhido, o selo vira um ponto sobre o ícone.
Defina Badge por código sempre que o valor mudar, a barra se repinta sozinha.

Exemplo

```
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;
```

Veja Também

- TsgcNavigationItem
- Temas
- Recolhimento e animação

Teclado e Foco

A barra recebe o foco do teclado (**TabStop** na VCL, **CanFocus** no FMX, ambos True por padrão) e move uma linha com foco entre os itens que podem receber foco: os itens exibidos e habilitados e, no accordion expandido, também os cabeçalhos de grupo.

Tecla	Ação
Up, Down	Move o foco para a linha anterior ou seguinte que pode receber foco e a rola para a área visível.
Home, End	Move o foco para a primeira ou a última linha que pode receber foco.
Enter, Space	Ativa a linha com foco como um clique: seleciona um item e dispara OnItemClick, ou alterna um grupo accordion.
Left, Right	Recolhe ou expande o grupo accordion com foco. Left em um item de um grupo move o foco para o cabeçalho do grupo.

O retângulo de foco tracejado só é exibido enquanto o teclado é usado. Seleção e foco são separados: **SelectedIndex** é a página que o usuário escolheu, **Model.FocusedIndex** é a linha em que o teclado está.

Dicas

Com **ShowHint** True, a dica é o **Hint** do item sob o cursor, ou sua Caption no trilho recolhido, de modo que ao passar para outra linha a dica dela é exibida. No FMX, as dicas requerem o 10.1 Berlin ou posterior.

Exemplo

```
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;
```

Veja Também

- TsgcVCLNavigationBar
- TsgcNavigationModel
- Modo accordion

Action Links

Um item se vincula a qualquer TBasicAction, uma TAction da VCL ou do FireMonkey, por meio de sua propriedade **Action**. Os Action links não fazem parte da edição Community, consulte Edições.

- **Caption**, **Enabled**, **Visible**, **Hint** e **ImageIndex** do item acompanham a action.
- Um clique no item dispara OnItemClick da barra e depois executa a action, com a barra como seu Action-Component. Quando o item tem um manipulador OnClick, o OnClick é executado no lugar da action.
- O OnUpdate das actions é executado quando o aplicativo está ocioso, porque a barra chama InitiateAction em cada item.
- No Delphi XE3 e posteriores, a action envia cada alteração ao item. Em versões mais antigas, o item copia as propriedades da action quando ela é atribuída e novamente a cada InitiateAction.

Exemplo

```
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;
```

Veja Também

- TsgcNavigationItem
- Demos
- Edições

Temas

Theme define as cores da barra a partir dos tokens de tema do sgcUICore: utLight (padrão), utDark, ou utCustom, que usa as cores do tema claro. A superfície, a borda na margem direita, o texto, os cabeçalhos de grupo esmaecidos, as sobreposições de passagem do mouse e de pressionado e o destaque vêm todos do tema.
AccentColor substitui o destaque do tema por uma cor $AARRGGBB, usada pela barra e pelo fundo da seleção, pelo retângulo de foco e pelos selos. 0 (padrão) mantém o destaque do tema.
Os tokens de tema são os mesmos do sgcDashboard e dos demais produtos sgcUICore, de modo que uma barra e um dashboard com o mesmo Theme combinam. Uma barra hospedada em um tile do sgcDashboard segue o tema do dashboard, porque o tile envia seu tema à propriedade publicada Theme dos controles que hospeda.

Exemplo

```
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;
```

Veja Também

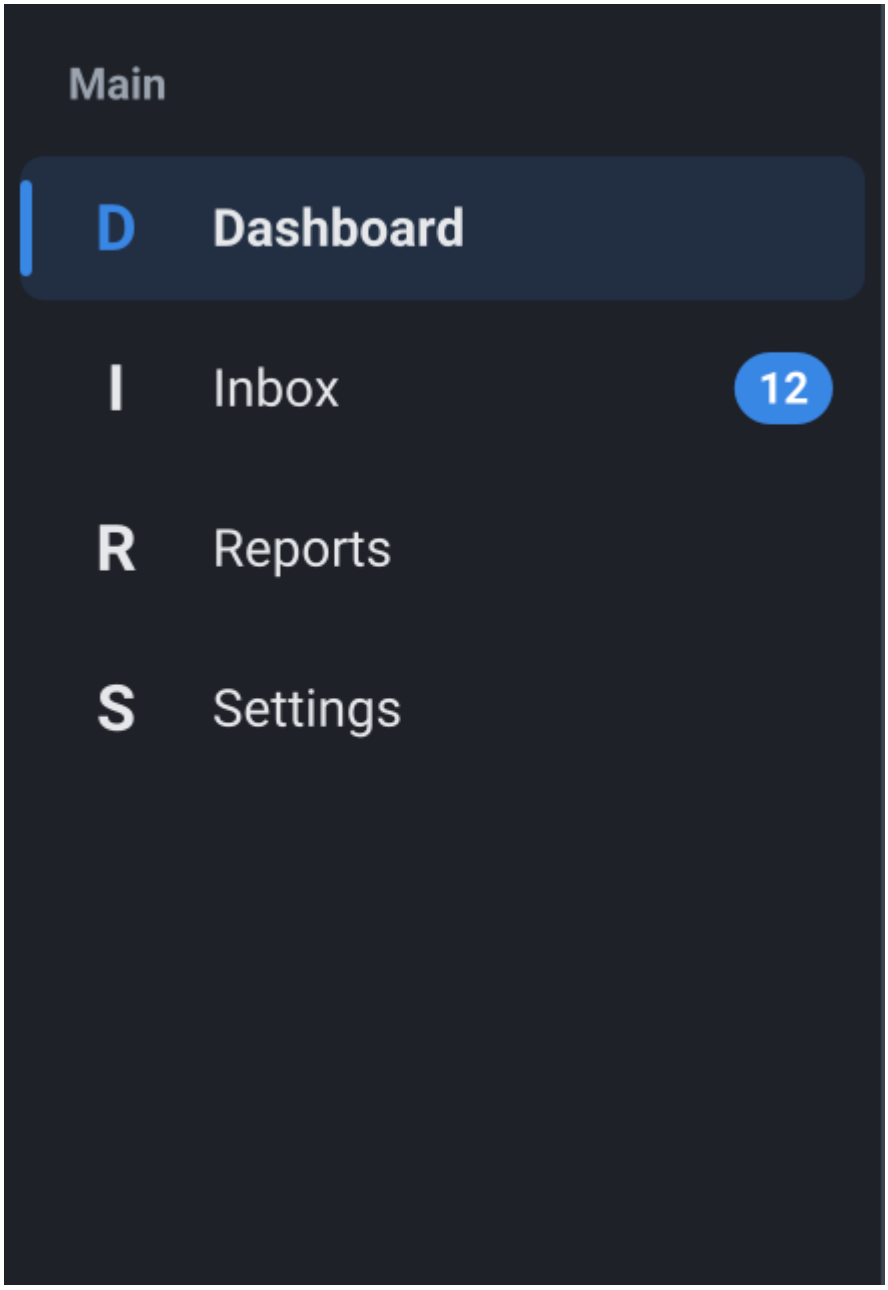
- Arquitetura
- Selos
- Renderização headless

Renderização Headless

O renderizador em sgcNavigation_Render desenha um TsgcNavigationModel sem janela e sem VCL nem FMX, de modo que um aplicativo de console, um serviço, um teste ou um servidor web pode produzir a mesma barra como imagem, por exemplo para a documentação de um aplicativo ou para um teste visual. A unit compila em todas as plataformas, inclusive Linux64.

Rotina	Descrição
sgcNavigationSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = nil)	Grava um arquivo ou stream PNG. aWidth e aHeight são pixels lógicos e aScale os multiplica: 220 por 320 na escala 2 gera uma imagem de 440 por 640.
sgcNavigationSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = nil)	Grava um arquivo ou stream SVG da mesma barra.
sgcNavigationRender(aModel, aCanvas, aWidth, aHeight, aScale = 1, aScrollY = 0, aImages = nil)	Desenha a barra em qualquer canvas do sgcUICore, em pixels do dispositivo.

aImages é um callback opcional do tipo TsgcNavigationImageEvent que retorna a imagem de um ImageIndex. Sem ele, os itens desenham seu Glyph. O modelo de um controle está disponível por sua propriedade **Model**, de modo que **sgcNavigationSaveToPNG(Nav.Model, ...)** salva o que o controle mostra em qualquer tamanho.



Exemplo

```
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;
```

Veja Também

- TsgcNavigationModel
- Plataformas

Visão Geral do .NET

sgcNavigation .NET é um port gerenciado em C# da biblioteca Delphi sgcNavigation, não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcNavigation** e mantêm os nomes do Delphi, de modo que **TsgcNavigationModel**, **TsgcNavigationItems**, **TsgcNavigationItem** e **TsgcNavigationMode** se leem igual nas duas linguagens, e a barra de navegação de cada host segue o padrão de nomes do Delphi. Ela é desenhada com o port .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual eSeGeCe, de modo que uma barra de navegação tem a mesma aparência no WinForms, no WPF, no Avalonia e na saída headless em PNG e SVG.
Uma barra de navegação mostra uma lista de itens com um ícone, uma legenda e um badge opcional. No modo sidebar todos os itens são exibidos e os cabeçalhos de grupo são rótulos estáticos. No modo accordion, um clique no cabeçalho de um grupo expande ou recolhe seus itens. Nos dois modos a barra se recolhe em um trilho de ícones, animado entre ExpandedWidth e CollapsedWidth.

Pacotes

O port é dividido em cinco pacotes NuGet, de modo que um projeto só traz o host que usa.

PackageId	Contém	Frameworks de destino
esegece.sgcNavigation	O modelo (**TsgcNavigationModel**, **TsgcNavigationItems**, **TsgcNavigationItem**), o vínculo **Command** do item, o renderizador **sgcNavigation_Render** e a saída headless em SVG. Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcNavigation.Rendering	A saída headless em PNG, **sgcNavigation_Rendering**, desenhada no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, no Linux e no 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**, no Windows, no Linux e no macOS com Avalonia 11.	net8.0, net9.0

Os três pacotes de host dependem de esegece.sgcNavigation.Rendering, que depende de esegece.sgcNavigation, de modo que uma linha `PackageReference` traz tudo o que um host precisa. Veja Instalação pelo NuGet.

Edições

Edição	IDs dos pacotes	Inclui

Registered	esegece.sgcNavigation, esegece.sgcNavigation.Rendering, esegece.sgcNavigation.WinForms, esegece.sgcNavigation.Wpf, esegece.sgcNavigation.Avalonia	Tudo: as barras de navegação dos três hosts, os modos sidebar e accordion, o trilho de ícones, imagens, badges, navegação pelo teclado, o vínculo Command do item, os temas e a saída headless em PNG e SVG.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcNavigation.WinForms.Community. Dependem dos pacotes Community do sgcUICore.	O mesmo conteúdo de Registered sem as propriedades **Command** e **CommandParameter** do item, pois a edição Community do Delphi não tem action links. Os itens são executados por meio do seu OnClick.

A edição Community é gratuita para uso comercial e não comercial por indivíduos e organizações com menos de EUR 1.000.000 de receita bruta anual e no máximo cinco desenvolvedores. Uma organização igual ou acima de qualquer um dos limites precisa de uma licença Registered. O port .NET não tem edição Trial.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	VCL (TsgcVCLNavigationBar)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	FMX (TsgcFMXNavigationBar)	Windows, .NET Framework 4.7.2 e .NET 8 e 9. A barra é declarada em XAML e os itens são adicionados no código.
Avalonia	FMX (TsgcFMXNavigationBar)	Windows, Linux e macOS, .NET 8 e 9, com mouse, toque e caneta. A barra é declarada em XAML e os itens são adicionados no código.
Headless	sgcNavigation_Render	Qualquer plataforma dos frameworks de destino: SVG com esegece.sgcNavigation, PNG com esegece.sgcNavigation.Rendering.

Todos os hosts têm a mesma API: Items, Mode, Collapsed, ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight, Indent, AllowMultiExpand, SelectedIndex, Theme, AccentColor, Images, AnimateCollapse, ScrollY, ItemAtPos, ItemRect, ScrollIntoView e os eventos OnItemClick, OnSelectionChange e OnCollapsedChange. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja Também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Renderização Headless no .NET
- Demos do .NET
- Plataformas

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, e os pacotes core e de renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcNavigation.WinForms
dotnet add package esegece.sgcNavigation.Wpf
dotnet add package esegece.sgcNavigation.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcNavigation.WinForms.Community
dotnet add package esegece.sgcNavigation.Wpf.Community
dotnet add package esegece.sgcNavigation.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcNavigation.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie os pacotes Registered ou os Community em um projeto, nunca os dois: eles contêm os mesmos tipos no mesmo namespace. O código que usa o **Command** do item não compila com os pacotes Community; proteja-o com `#if !SGC_COMMUNITY` quando um projeto compila as duas edições.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:sgc="clr-namespace:esegece.sgcNavigation;assembly=esegece.sgcNavigation.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicativo desktop Avalonia. Mapeie o namespace no XAML com `xmlns:sgc="using:esegece.sgcNavigation"`.
- Headless: esegece.sgcNavigation sozinho basta para SVG. Adicione esegece.sgcNavigation.Rendering para PNG, ele traz o SkiaSharp e seus recursos nativos para Windows e Linux.

Em C# todo host precisa de `using esegece.sgcNavigation;`, e de `using esegece.sgcUICore;` para **TsgcUITheme** e os demais tipos dos temas.

Veja Também

- Visão Geral do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Demos do .NET

Início Rápido WinForms

TsgcWinFormsNavigationBar é a barra de navegação do WinForms, a contrapartida do TsgcVCLNavigationBar do Delphi. Ela possui um **TsgcNavigationModel**, expõe suas propriedades e o desenha por meio do sgcUICore. O pacote é **esegece.sgcNavigation.WinForms**.

Passos

- Crie um projeto WinForms e adicione o pacote esegece.sgcNavigation.WinForms.
- Crie um **TsgcWinFormsNavigationBar**. Ele é acoplado à esquerda por padrão e toma sua largura de **ExpandedWidth**.
- Atribua um **ImageList** a **Images** e adicione os itens com `Items.AddItem(aCaption)`. Defina **ImageIndex**, **Badge** e **Hint** de cada item. `AddItem(aCaption, "", true)` adiciona um cabeçalho de grupo.
- Atribua **OnSelectionChange** para mostrar a página do item selecionado, e **OnItemClick** para reagir a cada clique.
- Execute: um clique ou Enter seleciona um item, as setas, Home e End movem o foco, e **Collapsed** recolhe a barra no trilho de ícones.

Exemplo

O Program.cs completo de um aplicativo WinForms com uma sidebar, um badge, um cabeçalho de grupo e um botão Collapse. Ele compila como está em um projeto `net8.0-windows` que referencia esegece.sgcNavigation.WinForms.
A barra também está na paleta de ferramentas. Os itens não são serializados pelo designer, por isso são sempre adicionados no código, veja Diferenças do .NET em relação ao Delphi.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WPF
- Início Rápido Avalonia
- Imagens e Glifos

Início Rápido WPF

TsgcWpfNavigationBar é a barra de navegação do WPF. Ela é declarada em XAML como qualquer controle, e seus itens são adicionados no código. O pacote é **esegece.sgcNavigation.Wpf**.

Passos

- Crie um projeto WPF (`dotnet new wpf`) e adicione o pacote esegece.sgcNavigation.Wpf.
- Mapeie o namespace `esegece.sgcNavigation` e declare um **TsgcWpfNavigationBar**. **Mode**, **AnimateCollapse**, **AllowMultiExpand** e as demais propriedades simples podem ser definidas como atributos XAML.
- No code behind atribua **Images**, uma lista de **ImageSource**, adicione os itens e atribua os eventos: eles são delegates, não eventos do WPF, portanto não podem ser definidos como atributos XAML.

Exemplo

MainWindow.xaml declara a barra, um botão Collapse e um texto que mostra a página selecionada. O App.xaml do projeto é o do template WPF, com StartupUri definido como MainWindow.xaml.
MainWindow.xaml.cs cria três ícones vetoriais **DrawingImage**, adiciona os itens e atribui os eventos.
Images aceita qualquer **ImageSource**, tanto um **BitmapImage** quanto um **DrawingImage**. Quando a lista é uma **ObservableCollection<ImageSource>**, a barra é redesenhada quando uma imagem é adicionada, removida ou substituída.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WinForms
- Início Rápido Avalonia
- Imagens e Glifos

Início Rápido Avalonia

TsgcAvaloniaNavigationBar é a barra de navegação do Avalonia, a contrapartida do TsgcFMXNavigationBar do Delphi, no Windows, no Linux e no macOS. Mouse, toque e caneta passam pelo mesmo caminho de ponteiro. O pacote é **esegece.sgcNavigation.Avalonia**.

Passos

- Crie um projeto .NET 8 e adicione os pacotes esegece.sgcNavigation.Avalonia, Avalonia.Desktop e Avalonia.Themes.Fluent.
- Mapeie o namespace com `xmlns:sgc="using:esegece.sgcNavigation"` e declare um **TsgcAvaloniaNavigationBar** com seu **Mode**.
- No code behind defina o **Theme**, adicione os grupos e os itens, e atribua os eventos: eles são delegates, não eventos do Avalonia.

Exemplo

MainWindow.axaml declara um accordion em que apenas um grupo fica aberto por vez.
MainWindow.axaml.cs define o tema escuro, adiciona três grupos com seus itens e mostra o item clicado.
Program.cs inicia o aplicativo com o tema Fluent.
Theme oculta a propriedade `StyledElement.Theme` dos controles Avalonia, por isso defina-o no código em uma variável do tipo TsgcAvaloniaNavigationBar, não como atributo XAML. Images é uma lista de **IImage**, um **Bitmap** ou um **DrawingImage**.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WinForms
- Início Rápido WPF
- Modo Accordion

Classes do .NET

Cada host tem uma classe de barra de navegação com os mesmos membros. Os tipos neutros de framework são compartilhados pelos três hosts e mantêm seus nomes do Delphi.

Classes de host

Host	Classe	Classe base	Delphi
WinForms	TsgcWinFormsNavigationBar	TsgcUIWinFormsControl	TsgcVCLNavigationBar
WPF	TsgcWpfNavigationBar	TsgcUIWpfControl	TsgcFMXNavigationBar
Avalonia	TsgcAvaloniaNavigationBar	TsgcUIAvaloniaControl	TsgcFMXNavigationBar

Tipos neutros de framework

Tipo	Descrição
TsgcNavigationModel	Os itens, o modo, o trilho recolhido, os itens selecionado, sob o mouse, pressionado e com foco, o layout das linhas e o hit test. Cada barra possui um, veja TsgcNavigationModel.
TsgcNavigationItems	A coleção de itens: Count, o indexador, Add, AddItem(aCaption, aGlyph = "", aIsGroup = false), Insert, Delete, Clear, IndexOf, BeginUpdate e EndUpdate.
TsgcNavigationItem	Um item: Caption, Glyph, ImageIndex, Badge, IsGroup, Expanded, Enabled, Visible, Hint, Tag (long), OnClick, Click() e, na edição Registered, Command e CommandParameter. Veja TsgcNavigationItem.
TsgcNavigationMode	nbmSidebar, nbmAccordion.
TsgcNavigationPart	npNone, npItem, npExpander: a parte de uma linha retornada por ItemAtPos.
TsgcNavigationItemEvent	`void (object Sender, TsgcNavigationItem aItem)`, o tipo de OnItemClick.
TsgcUINotifyEvent	`void (object Sender)`, de esegece.sgcUICore.
sgcNavigation_Render, sgcNavigation_Rendering	O renderizador e a saída headless em SVG e PNG, veja Renderização Headless no .NET.

Barra de navegação

Membro	Tipo	Descrição
Model	TsgcNavigationModel	O modelo da barra.
Items	TsgcNavigationItems	Os itens e os cabeçalhos de grupo, em ordem.
Mode	TsgcNavigationMode	Sidebar ou accordion.

Collapsed	bool	Recolhe a barra no trilho de ícones. A largura passa de ExpandedWidth para CollapsedWidth, com animação quando AnimateCollapse é true.
ExpandedWidth, CollapsedWidth	int	A largura da barra expandida e do trilho de ícones, 220 e 56 por padrão.
ItemHeight, GroupHeight, Indent	int	A altura de uma linha de item, de uma linha de cabeçalho de grupo, e o recuo dos itens de um grupo no modo accordion.
AllowMultiExpand	bool	No modo accordion, false fecha os outros grupos quando um se abre.
SelectedIndex	int	O item selecionado, ou -1.
Theme, AccentColor	TsgcUITheme, uint	O tema da barra. AccentColor 0 usa a cor de destaque do tema.
Images	ImageList, IList<ImageSource>, IList<IImage>	As imagens de ImageIndex, veja Diferenças do .NET em relação ao Delphi.
AnimateCollapse	bool	Anima Collapsed durante cerca de 150 ms.
ScrollY	int	O deslocamento vertical da rolagem em pixels do dispositivo.
ItemAtPos(X, Y), ItemAtPos(X, Y, out aPart)	int	O índice do item em um ponto do controle, ou -1, e a parte da linha.
ItemRect(aIndex)	Rectangle, Rect	A linha de um item nas coordenadas do controle.
ScrollIntoView(aIndex)	void	Rola a barra para que a linha do item fique visível.
OnItemClick	TsgcNavigationItemEvent	Um clique, Enter ou Espaço em um item habilitado, antes que seu OnClick ou Command seja executado. Um cabeçalho de grupo não o dispara.
OnSelectionChange	TsgcUINotifyEvent	SelectedIndex mudou, pelo usuário ou no código.
OnCollapsedChange	TsgcUINotifyEvent	Collapsed mudou.

As posições e os tamanhos são pixels no WinForms (Rectangle) e unidades independentes de dispositivo no WPF e no Avalonia (Rect).

Veja Também

- Visão Geral do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia

Diferenças do .NET em relação ao Delphi

A API .NET segue a API Delphi nome por nome. Estes são os pontos em que ela difere.

Command em vez de action links

O Delphi vincula um item a um TBasicAction por meio da sua propriedade Action, veja Action Links. O .NET não tem TBasicAction, por isso o item tem **Command**, um `System.Windows.Input.ICommand`, e **CommandParameter**. Ambos fazem parte apenas da edição Registered, como os action links do Delphi.

- `CanExecute(CommandParameter)` controla o **Enabled** do item: quando o comando ou o parâmetro é atribuído e a cada `CanExecuteChanged`.
- Um clique, Enter ou Espaço dispara OnItemClick da barra e depois executa o comando, quando ele pode ser executado. Quando o item tem um manipulador OnClick, o OnClick é executado no lugar do comando, como no Delphi.
- Caption, Hint e ImageIndex não são sincronizados a partir do comando, porque o ICommand não carrega nenhum deles. Continuam sendo propriedades do item. Visible também não é sincronizado.
- Não há OnUpdate em ociosidade. Um comando que muda seu estado dispara CanExecuteChanged, e o item o acompanha imediatamente.

Um programa de console que vincula um item a um comando mínimo, o executa e o desabilita por meio de CanExecuteChanged. Ele precisa apenas de esegece.sgcNavigation, as barras dos três hosts se comportam da mesma forma.

Images

O Delphi usa um TCustomImageList no VCL e um TCustomImageList de FMX.ImgList no FireMonkey, veja Imagens e Glifos. No .NET a propriedade **Images** tem o tipo de imagem de cada host, e o **ImageIndex** do item é um índice nela.

Host	Images	Desenhada em
WinForms	System.Windows.Forms.ImageList	O ImageSize da lista de imagens, em pixels.
WPF	IList<ImageSource>: bitmaps e ícones vetoriais DrawingImage. Uma ObservableCollection é redesenhada quando muda.	16 x 16 unidades independentes de dispositivo, nítida em qualquer DPI para ícones vetoriais.
Avalonia	IList<IImage>: um Bitmap ou um DrawingImage. Uma lista com INotifyCollectionChanged é redesenhada quando muda.	16 x 16 unidades independentes de dispositivo.

Cada imagem é convertida uma vez para ARGB e mantida em cache até a lista mudar e, no WPF e no Avalonia, também até a escala de DPI mudar. Quando um item não tem imagem, o texto do seu **Glyph** é desenhado no espaço do ícone.

Dicas e tooltips

No Delphi a barra mostra o Hint do item sob o cursor quando ShowHint é True, e o FMX mostra dicas a partir do 10.1 Berlin. No .NET a barra sempre mostra um tooltip, sem propriedade ShowHint: o Hint do item sob o cursor, ou seu Caption no trilho recolhido, onde a legenda não está visível. O WinForms usa um componente ToolTip interno, o WPF define a propriedade ToolTip da barra e o Avalonia define a propriedade anexada ToolTip.Tip. Não defina você mesmo o tooltip da barra, a barra o substitui; defina o Hint do item.

Unidades

- WinForms: ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight e Indent são pixels lógicos multiplicados por DPI / 96, como a barra VCL com PPI / 96. ItemAtPos, ItemRect e ScrollY são pixels da área cliente.
- WPF e Avalonia: Width, ExpandedWidth, CollapsedWidth, ItemAtPos e ItemRect são unidades independentes de dispositivo, como no FMX. O quadro, o layout do modelo e **ScrollY** são pixels do dispositivo, as unidades multiplicadas pela escala de DPI do controle (WPF) ou pela escala de renderização da janela (Avalonia). O BoundsRect de um item também está em pixels do dispositivo.
- Headless: aWidth e aHeight são pixels lógicos e aScale os multiplica, veja Renderização Headless no .NET.

Eventos

OnItemClick, OnSelectionChange, OnCollapsedChange e o OnClick do item são propriedades delegate com os nomes do Delphi, não membros `event` do C#. Atribuir um manipulador substitui o anterior, como um ponteiro de método do Delphi, e eles são atribuídos no código, também no WPF e no Avalonia, onde o XAML não pode defini-los.

Sem verbos de designer

No Delphi a propriedade Items é editada em tempo de design com o editor de coleção padrão. O port .NET não tem verbos de designer nem editor de coleção: os Items de uma barra não são serializados, por isso são sempre adicionados no código, geralmente no construtor do formulário ou da janela. No WinForms a barra está na paleta de ferramentas e suas propriedades simples são definidas na janela Propriedades. No WPF e no Avalonia a barra é declarada em XAML.

Edições

O port .NET é distribuído como pacotes Registered e Community, veja Visão Geral do .NET. Os pacotes Community não têm Command nem CommandParameter. Não há edição Trial.

Vinculação de dados

No WPF e no Avalonia, as propriedades visíveis ao usuário são propriedades de dependência e propriedades do Avalonia, portanto podem ser vinculadas, estilizadas e animadas, veja Vinculação de dados no .NET.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Action Links
- Imagens e Glifos

Vinculação de dados no .NET

Nos hosts WPF e Avalonia, as propriedades visíveis ao usuário da barra de navegação são propriedades de dependência (WPF) e propriedades do Avalonia, portanto podem ser vinculadas a um view model, definidas por um Style e animadas. Este tópico se aplica somente a WPF e Avalonia. O controle WinForms não muda: suas propriedades são propriedades CLR simples, como antes.

Propriedades vinculáveis

Cada propriedade da tabela tem um campo estático com o sufixo Property, por exemplo `TsgcWpfNavigationBar.SelectedIndexProperty`. No WPF os campos são DependencyProperty. No Avalonia são StyledProperty, exceto SelectedIndex, que é uma DirectProperty. As propriedades marcadas com **TwoWay** são vinculadas em TwoWay por padrão, porque a barra também as altera (SelectedIndex após um clique ou uma tecla, ambas após uma alteração no modelo), de modo que uma vinculação sem Mode mantém o view model atualizado. As demais propriedades são vinculadas em OneWay por padrão.

Classe WPF	Classe Avalonia	Propriedades vinculáveis
TsgcWpfNavigationBar	TsgcAvaloniaNavigationBar	Mode, **Collapsed (TwoWay)**, ExpandedWidth, CollapsedWidth, ItemHeight, GroupHeight, Indent, AllowMultiExpand, **SelectedIndex (TwoWay)**, Theme, AccentColor, AnimateCollapse

No Avalonia, Theme é o tema do sgcUICore (TsgcUITheme) e oculta o tema de controle do Avalonia de StyledElement, que continua acessível por meio de `StyledElement.ThemeProperty`.

Comportamento

- **O modelo mantém os valores.** Cada propriedade é armazenada no modelo de navegação. Um valor vinculado vai para o modelo, e a propriedade retoma o valor que o modelo mantém: o modelo rejeita uma largura ou uma altura menor que 1 e mantém SelectedIndex dentro dos itens, e a propriedade então mostra o valor em uso. Uma origem TwoWay também recebe o valor corrigido, no WPF e no Avalonia. No WPF ele chega depois que o dispatcher executa o trabalho pendente. No Avalonia a gravação de volta cobre um valor que a origem envia quando a vinculação já está em execução. Um valor que a origem já tem quando a vinculação inicia é corrigido no controle, mas não é gravado de volta na origem.
- **As alterações feitas pela barra chegam ao view model.** Um clique em um item, uma tecla, uma alteração no modelo e uma alteração pelo código atualizam SelectedIndex, e uma alteração no modelo ou pelo código atualiza Collapsed. Com a vinculação TwoWay padrão, o view model acompanha.
- **Os eventos disparam para qualquer origem.** OnSelectionChange e OnCollapsedChange disparam para uma alteração por entrada do usuário, pelo código ou por uma vinculação.
- **Primeiro os itens, depois a vinculação.** SelectedIndex se refere a um item, portanto adicione os itens antes que o valor vinculado chegue, por exemplo definindo o DataContext depois dos itens, como fazem os exemplos.
- **Styles.** Um setter de Style pode definir qualquer propriedade vinculável, por exemplo Mode e Theme de todas as barras de navegação de uma janela.

Propriedades que continuam sendo propriedades CLR

Propriedade	Motivo
Items	A árvore de itens do modelo, construída com AddItem e os demais métodos da coleção, e não um valor único para vincular.
Images	A lista de imagens dos itens, um objeto compartilhado por referência.

Model	O modelo independente de plataforma por trás do controle, somente leitura.
ScrollY	A posição de rolagem, que muda a cada passo da roda e não é uma configuração.
OnSelectionChange, OnCollapsedChange e os demais eventos	Propriedades delegate com os nomes do Delphi, atribuídas no código, veja Diferenças do .NET em relação ao Delphi.

Exemplo

O view model abaixo é compartilhado pelos dois exemplos. SelectedIndex e Collapsed estão vinculados sem Mode, portanto são vinculados em TwoWay: um clique em um item altera o view model, e a caixa de seleção recolhe a barra por meio do view model.

WPF

MainWindow.xaml vincula a barra. App.xaml é o do modelo WPF.
MainWindow.xaml.cs adiciona os itens e depois define o view model como DataContext.

Avalonia

MainWindow.axaml define `x:DataType` como o view model, de modo que as vinculações são vinculações compiladas. A classe App é a de Início Rápido Avalonia.
MainWindow.axaml.cs adiciona os itens e depois define o view model como DataContext.

Veja Também

- Diferenças do .NET em relação ao Delphi
- Classes do .NET
- Início Rápido WPF
- Início Rápido Avalonia

Renderização Headless no .NET

Uma barra de navegação pode ser desenhada sem janela, em um servidor, em um teste ou em um relatório. **sgcNavigation_Render.sgcNavigationSaveToSVG** grava um arquivo ou stream SVG apenas com o pacote esegece.sgcNavigation, e **sgcNavigation_Rendering.sgcNavigationSaveToPNG** grava um PNG com esegece.sgcNavigation.Rendering, que desenha no SkiaSharp no Windows, no Linux e no macOS. A saída SVG é idêntica byte a byte à da renderização headless do Delphi.

Métodos

Método	Pacote	Descrição
sgcNavigation_Render.sgcNavigationSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = null)	esegece.sgcNavigation	Grava a barra de aModel como SVG.
sgcNavigation_Rendering.sgcNavigationSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1, aImages = null)	esegece.sgcNavigation.Rendering	Grava a barra de aModel como PNG com fundo transparente.
sgcNavigation_Render.sgcNavigationRender(aModel, aCanvas, aWidth, aHeight, aScale = 1, aScrollY = 0, aImages = null)	esegece.sgcNavigation	Desenha a barra em qualquer canvas do sgcUICore.

aWidth e aHeight são pixels lógicos. A imagem tem aWidth x aScale por aHeight x aScale pixels, de modo que uma escala 2 dá uma imagem nítida para uma tela de alto DPI ou para impressão. aImages é um **TsgcNavigationImageEvent** opcional, `bool (int aImageIndex, out TsgcUIARGBBuffer aImage)`, que retorna a imagem de um ImageIndex como um **TsgcUIARGBBuffer**; sem ele o Glyph do item é desenhado no espaço do ícone.

Exemplo

Um programa de console (`net8.0`) que referencia esegece.sgcNavigation.Rendering e grava em PNG e SVG uma sidebar com um badge, um cabeçalho de grupo e um item desabilitado, o trilho de ícones recolhido, e os mesmos itens como um accordion escuro com um grupo fechado.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Renderização Headless
- TsgcNavigationModel

Demos do .NET

As demos .NET estão na pasta `demos` do produto e no zip de demos, com a solução `sgcNavigation_Demos.sln`. Cada demo tem um README.md.

Demo	Host	Descrição
SidebarApp	WinForms	Uma sidebar à esquerda alterna as páginas de um aplicativo. Os cinco itens (Dashboard, Inbox, Reports, Billing e Settings) tiram seus ícones de um ImageList, Inbox e Reports mostram um badge e Billing está desabilitado. O botão Collapse recolhe a barra no trilho de ícones com uma animação. OnSelectionChange mostra a página guardada no Tag do item selecionado, OnItemClick escreve o item clicado na linha de status e OnCollapsedChange atualiza a legenda do botão. A primeira página hospeda um dashboard sgcDashboard com dois cartões KPI ao vivo, e o combo Theme alterna a barra, o dashboard e as páginas entre claro e escuro.
SidebarAppWpf	WPF	O mesmo aplicativo em um TsgcWpfNavigationBar com ícones vetoriais DrawingImage e um TsgcWpfDashboard declarado em XAML.
SidebarAppAvalonia	Avalonia	O mesmo aplicativo em um TsgcAvaloniaNavigationBar com ícones vetoriais IImage, no Windows, no Linux e no macOS.
Accordion	WinForms	Um accordion com os grupos Sales, Inventory e Admin. Com "Allow multi expand" desmarcado, abrir um grupo fecha os outros. New order, Export e Audit log estão vinculados a comandos por meio do Command do item: um clique executa o comando e o item Export acompanha CanExecute, que a caixa de seleção "Export command enabled" altera. O botão Rename renomeia o próprio item, porque o ICommand não carrega legenda. Na edição Community os itens são executados por meio do seu OnClick.
AccordionWpf	WPF	O mesmo accordion em um TsgcWpfNavigationBar.
AccordionAvalonia	Avalonia	O mesmo accordion em um TsgcAvaloniaNavigationBar.

Compilar e executar

As demos precisam do SDK do .NET 8 ou posterior. Execute uma demo a partir da sua pasta:

```
dotnet run
```

Dentro da árvore do produto as demos são compiladas contra os projetos da biblioteca. A partir do zip de demos elas restauram os pacotes NuGet, a edição Community por padrão. Defina **SgcNavigationEdition** para usar os pacotes Registered:

```
dotnet build -p:SgcNavigationEdition=Registered
```

Autoverificação

Cada demo tem uma opção `--test` que abre a janela fora da tela, clica nos itens, abre e fecha os grupos, recolhe a barra e alterna o tema, grava uma linha PASS ou FAIL por verificação no arquivo e termina com o número de verificações que falharam, de modo que 0 significa que todas passaram.

```
dotnet run -- --test result.txt
```

Veja Também

- Visão Geral do .NET
- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Demos

Solução de Problemas

Undeclared identifier Action, ou "Property Action does not exist"

A edição Community compila sem SGC_NAVIGATION_ACTIONS, portanto a propriedade Action dos itens não existe. O código que a usa falha com E2003 Undeclared identifier, e um formulário salvo por uma compilação Registered com um item vinculado a uma action gera uma exceção "Error reading" ao ser carregado. Use o OnClick do item ou o evento OnItemClick da barra, ou uma compilação Registered ou Trial. Consulte Edições.

Um clique em um cabeçalho de grupo não o seleciona

Os cabeçalhos de grupo nunca são selecionados. No modo sidebar eles são rótulos estáticos, e no accordion expandido um clique alterna o grupo. Use um item com IsGroup False para uma página. Consulte Modo accordion.

Um item não reage a um clique

Um item desabilitado (Enabled False) é desenhado esmaecido e ignora cliques e o teclado. Um item com uma Action segue o Enabled da action, portanto verifique também o OnUpdate da action.

As imagens não aparecem no FireMonkey

A propriedade Images da barra FMX requer o Delphi XE8 ou posterior. Em versões mais antigas, os itens desenham seu Glyph. Consulte Imagens e glifos.

Sem dicas na barra FMX

As dicas dos itens da barra FMX requerem o Delphi 10.1 Berlin ou posterior, e ShowHint deve ser True.

A barra salta para a nova largura em vez de animar

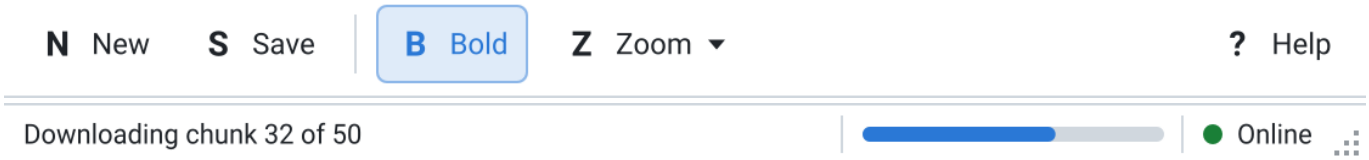
A animação de recolhimento só é executada em tempo de execução e apenas enquanto AnimateCollapse é True. Em tempo de design, a largura muda de uma vez.

A demo SidebarApp, SidebarFMX ou Accordion não compila

SidebarApp e SidebarFMX hospedam um dashboard e requerem o sgcDashboard instalado. Accordion usa Action links, que não fazem parte da edição Community. Consulte Demos.

Introdução

sgcToolbar é uma barra de ferramentas e uma barra de status nativas para Delphi e C++Builder, para VCL e FireMonkey. **TsgcVCLToolbar** e **TsgcFMXToolbar** mostram uma fileira de botões, alternadores, grupos de rádio, botões suspensos, rótulos, separadores e espaçadores, e movem os itens que não cabem para um menu de excedentes. **TsgcVCLStatusBar** e **TsgcFMXStatusBar** mostram painéis de texto, barras de progresso e indicadores de estado coloridos, e uma thread de trabalho pode atualizá-los diretamente por meio de setters thread-safe.
O sgcToolbar faz parte da suíte visual da eSeGeCe e dá aos outros produtos a sua moldura: uma barra de ferramentas na parte superior e uma barra de status na parte inferior de uma janela que hospeda um painel **sgcDashboard**, uma grade **sgcGrid** ou uma barra de navegação **sgcNavigation**. Ele é desenhado com o **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro, de modo que as barras têm a mesma aparência no VCL, no FireMonkey e na saída headless em PNG e SVG. O sgcToolbar requer o sgcUICore, que é instalado junto com ele.



Edições

O sgcToolbar é distribuído nas edições Registered, Trial e Community. A edição Community tem todos os recursos, exceto o menu de excedentes e os Action links dos itens. Consulte Edições.

Por onde começar

- Arquitetura: os modelos independentes de framework, o renderizador sobre o canvas do sgcUICore e os controles VCL e FMX.
- Plataformas: as versões do Delphi, o C++Builder e as plataformas.
- Início rápido VCL, Início rápido FMX e Início rápido C++Builder: uma primeira barra de ferramentas e barra de status em poucos minutos.
- Componentes: as barras de ferramentas, as barras de status, os itens, os painéis e os modelos.
- Tipos de item, Alternadores e grupos de rádio, Menus suspensos, Excedentes, Action links, Imagens e glifos e Teclado.
- Painéis de status, Progresso e indicadores, Atualizações thread-safe e Alça de redimensionamento.
- Temas e Renderização headless.
- Demos: os projetos de demonstração em Delphi e C++Builder.

Arquitetura

O sgcToolbar é construído em camadas. Cada camada depende apenas das que estão abaixo dela, de modo que os modelos e o renderizador compilam sem VCL ou FireMonkey.

Camada	Unit	Função
Modelos	sgcToolbar_Model	TsgcToolbarModel contém os Items (uma coleção de TsgcToolbarItem), ShowCaptions, os tamanhos, o tema e o estado de interação (item sob o mouse, pressionado e com foco). TsgcStatusBarModel contém os Panels (uma coleção de TsgcStatusBarPanel), o tema e a alça de redimensionamento. Seus métodos Layout calculam o retângulo de cada item ou painel, e HitTest encontra o que está sob um ponto.
Renderizador	sgcToolbar_Render	RenderToolbar e RenderStatusBar desenham um modelo em qualquer canvas do sgcUICore. Os métodos SaveToPNG e SaveToSVG dos modelos os usam para gravar uma imagem sem janela. Consulte Renderização headless.
Controles VCL	sgcToolbar_VCL	TsgcVCLToolbar e TsgcVCLStatusBar possuem um modelo, publicam suas propriedades e o desenham por meio da base de controles VCL do sgcUICore. Eles acrescentam o mouse, o teclado, as dicas, a lista de imagens, os menus popup e a alça de redimensionamento.
Controles FMX	sgcToolbar_FMX	TsgcFMXToolbar e TsgcFMXStatusBar são os controles FireMonkey com as mesmas propriedades, métodos e eventos, além de entrada por toque.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme	O runtime compartilhado: o canvas vetorial e o rasterizador, a base de controles VCL e FMX com controle de quadros (MaxFPS) e os tokens dos temas claro e escuro.

Modelo e controle

As propriedades publicadas de uma barra são as propriedades do seu modelo, e a propriedade **Model** da barra dá acesso ao restante: Layout, HitTest, a lista de excedentes e o estado de interação. Uma alteração no modelo agenda um novo quadro e nunca desenha diretamente, de modo que muitas alterações seguidas custam uma única repintura.

Desenho

Cada item e painel é desenhado pelo renderizador: o espaço do ícone com uma imagem ou um glifo, o texto, a seta do menu suspenso, os estados sob o mouse, pressionado, ativado, com foco e desabilitado, os separadores, o botão de excedentes, as barras de progresso, os pontos indicadores e a alça de redimensionamento. Todos os tamanhos são pixels lógicos multiplicados pela escala do monitor, de modo que as barras ficam nítidas em qualquer DPI. Os controles VCL usam PPI / 96 como escala e os controles FMX usam a escala do formulário.

Altura

A altura de uma barra de ferramentas acompanha seus botões: ButtonSize mais duas vezes Padding, em pixels lógicos, e qualquer outra altura é ignorada. Uma barra de status mantém o Height que você definir, 24 pixels lógicos por padrão.

Veja também

- Plataformas
- TsgcToolbarModel
- TsgcStatusBarModel
- Temas

Edições

O sgcToolbar é compilado a partir do mesmo código-fonte para todas as edições. A edição altera apenas quais recursos uma compilação inclui.

Edição	Inclui
Registered	Tudo: as barras de ferramentas e de status VCL e FMX, todos os tipos de item, alternadores e grupos de rádio, menus suspensos, o menu de excedentes, os Action links dos itens, imagens e glifos, navegação pelo teclado, painéis de texto, progresso e indicador, os setters thread-safe, a alça de redimensionamento, os temas e a saída headless em PNG e SVG.
Trial	Os mesmos recursos da Registered. Uma compilação feita com SGC_TRIAL exibe um aviso de versão de avaliação depois que a janela de lançamento indicada na compilação tiver passado. Nada mais muda.
Community	Tudo, exceto o menu de excedentes e os Action links dos itens.

O que a Community deixa de fora

A edição Community é compilada com SGC_TOOLBAR_ADVANCED indefinido. Em uma compilação Community:

- A barra de ferramentas não tem botão de excedentes. Os itens que não cabem são cortados na borda direita, e a propriedade **OverflowMenu** das barras de ferramentas não existe. Consulte Excedentes.
- A propriedade **Action** de TsgcToolbarItem não existe, portanto um item não pode ser vinculado a um TAction. Em vez disso, atribua o **OnClick** do item ou o evento **OnItemClick** da barra de ferramentas. Consulte Action links.

A edição é uma escolha em tempo de compilação no sgcVer.inc (SGC_COMMUNITY, que também desativa SGC_TRIAL), e não uma chave em tempo de execução. Um código ou um arquivo de formulário que use Action ou OverflowMenu não compila nem carrega em uma compilação Community, consulte Solução de problemas.

Veja também

- Excedentes
- Action links
- Plataformas

Instalação

O sgcToolbar faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcToolbar** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcToolbar

sgcUICore, a base compartilhada dos produtos de layout e do sgcCharts, acompanha o sgcToolbar. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcToolbar é selecionado. O sgcToolbar não tem pacote de ponte sgcWebSockets.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13`.
`Demos\Toolbar`	Os demos do sgcToolbar. Veja Demos.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcToolbar.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcToolbar. Os seus produtos são sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard, enquanto o sgcPDF e o sgcMaps não fazem parte do Community. Veja Edições para saber o que a compilação Community do sgcToolbar deixa de fora.

Instalações autônomas anteriores

Se o instalador encontrar uma instalação autônoma mais antiga do sgcUICore, como a gravada pelo instalador autônomo do sgcCharts, ele lista os library paths e os pacotes de design dessa instalação na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcToolbarD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcToolbar<V>` / `dclsgcToolbar<V>`	Os componentes do sgcToolbar (runtime / design time).

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcToolbar`, depois `dclsgcToolbar`. Em seguida, instale o pacote de design time `dclsgcToolbar` e adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcToolbarC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

.NET

Nada muda para o .NET. O sgcToolbar continua sendo distribuído como seus próprios pacotes NuGet (`esegece.sgcToolbar` e os pacotes complementares). Veja a instalação do .NET.

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	TsgcVCLToolbar e TsgcVCLStatusBar no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	TsgcFMXToolbar e TsgcFMXStatusBar a partir do Delphi XE6 (SGC_TOOLBAR_FMX), em Win32, Win64, macOS, iOS e Android, e no Linux com o FMX para Linux (FMXLinux). A propriedade Images requer o XE8 ou posterior (FMX.ImgList), e as dicas requerem o 10.1 Berlin ou posterior.
Linux64	As units independentes de framework, os modelos e o renderizador com a saída headless em PNG e SVG, a partir do Delphi 10.2 (SGC_TOOLBAR_LINUX).
.NET	Um port em C# com a mesma API: TsgcWinFormsToolbar e TsgcWinFormsStatusBar, TsgcWpfToolbar e TsgcWpfStatusBar, TsgcAvaloniaToolbar e TsgcAvaloniaStatusBar, e a saída headless em PNG e SVG, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão geral do .NET.

Pacotes

O sgcToolbar traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcToolbarD13.dpk para o Delphi 13 e sgcToolbarC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcToolbar	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, sgcUICore
sgcToolbarVCL	Runtime, VCL	sgcToolbar_VCL (TsgcVCLToolbar, TsgcVCLStatusBar)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcToolbar
sgcToolbarFMX	Runtime, FireMonkey, a partir do XE6	sgcToolbar_FMX (TsgcFMXToolbar, TsgcFMXStatusBar)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcToolbar
dclsgcToolbar	Tempo de projeto, VCL	Registra TsgcVCLToolbar e TsgcVCLStatusBar na página **SGC Toolbar** da paleta.	designide, rtl, vcl, sgcToolbar, sgcToolbarVCL
dclsgcToolbarFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXToolbar e TsgcFMXStatusBar na página **SGC Toolbar** da paleta.	designide, rtl, fmx, sgcToolbar, sgcToolbarFMX, dclsgcToolbar

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcToolbarFMX exige o dclsgcToolbar, portanto instale primeiro o dclsgcToolbar.

Versões do IDE e plataformas de destino

Delphi	sgcToolbar	sgcToolbarVCL	sgcToolbarFMX	dclsgcToolbar, dclsgcToolbarFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcToolbar, sgcToolbarVCL	sgcToolbarFMX	dclsgcToolbar, dclsgcToolbarFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcToolbar)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcToolbar)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcToolbarC13.cbproj, sgcToolbarVCLC13.cbproj, sgcToolbarFMXC13.cbproj, dclsgcToolbarC13.cbproj e dclsgcToolbarFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXToolbar e TsgcFMXStatusBar) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcToolbarVCL funcionam apenas no Windows.

- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcToolbar_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcToolbar é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa as unidades independentes de framework. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.
As barras VCL só aparecem na paleta de um formulário VCL e as barras FMX só na paleta de um formulário FireMonkey. As propriedades **Items** e **Panels** são editadas com o editor de coleções padrão.

Veja também

- Edições
- Arquitetura
- Solução de problemas

Início rápido VCL

Em tempo de design

- Solte um **TsgcVCLToolbar**, da página **SGC Toolbar** da paleta, em um formulário. Por padrão ele é alinhado ao topo.
- Dê um duplo clique em **Items** no Object Inspector e adicione os itens. Defina **Caption** e **Glyph** (uma letra ou um símbolo curto) de cada botão e defina **Kind** como tikSeparator, tikToggle, tikDropdown, tikLabel ou tikSpacer onde precisar de um deles.
- Solte um **TsgcVCLStatusBar**. Por padrão ele é alinhado à base. Dê um duplo clique em **Panels** e adicione os painéis, com **Spring** definido como True no painel que ocupa a largura livre.
- Escreva um manipulador **OnItemClick** para a barra de ferramentas ou atribua o **OnClick** de cada item.
- Defina **Theme** como utDark nas duas barras para usar a paleta escura.

No código

As mesmas barras criadas em tempo de execução. O programa compila como está com `sgcUI\Source` no caminho de busca.

```
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;
```

Veja também

- TsgcVCLToolbar
- TsgcVCLStatusBar
- TsgcToolbarItem
- Tipos de item
- Painéis de status

Início rápido FMX

Os controles FireMonkey requerem o Delphi XE6 ou posterior. Eles são usados exatamente como os controles VCL: solte um **TsgcFMXToolbar**, que por padrão é alinhado ao topo, e um **TsgcFMXStatusBar**, alinhado à base, preencha **Items** e **Panels** com o editor de coleções e trate o **OnItemClick**. Um toque em um botão funciona como um clique.

No 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;
```

Veja também

- TsgcFMXToolbar
- TsgcFMXStatusBar
- Plataformas

Início rápido C++Builder

Instale os pacotes do sgcToolbar para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página **SGC Toolbar** da paleta exatamente como no Delphi. O instalador adiciona os dois caminhos à IDE para você. Os cabeçalhos .hpp e os arquivos .lib gerados do sgcToolbar e do sgcUICore estão nessa única pasta, portanto o sgcUICore não precisa de um caminho próprio.

No código

OnItemClick é um evento de closure, portanto seu manipulador é uma função membro __fastcall de uma classe.

Veja também

- Início rápido VCL
- Demos
- Plataformas

Demos

Os demos estão na pasta **Demos** da instalação, com os projetos Delphi em Demos\Delphi e as versões C++Builder dos demos VCL em Demos\CBuilder. Cada pasta de demo tem um readme.html que explica o que o demo faz e como funciona.

Demo	Componentes	O que mostra
EditorShell	TsgcVCLToolbar, TsgcVCLStatusBar, TActionList	A estrutura de um pequeno editor de texto: botões Novo, Abrir e Salvar vinculados a ações, os alternadores Negrito e Itálico, um grupo de rádio Esquerda, Centro e Direita, um menu suspenso de Zoom, um espaçador, um alternador de tema escuro e um menu de excedentes, e uma barra de status com a posição do cursor e um indicador de estado de salvamento. Delphi e C++Builder, Win32 e Win64. Os Action links e o menu de excedentes não fazem parte da edição Community, portanto este demo não compila com ela.
EditorShellFMX	TsgcFMXToolbar, TsgcFMXStatusBar, TActionList	O demo EditorShell para FireMonkey, com os mesmos itens, ações e painéis. Delphi, Win32 e Win64. Ele também requer uma compilação Registered ou Trial.
LiveStatus	TsgcVCLStatusBar, TsgcVCLToolbar, TThread	Uma thread de trabalho simula um download e atualiza diretamente um painel de texto elástico, um painel de progresso, um painel indicador e um contador AutoSize com SetPanelText, SetProgress e SetIndicator, sem Synchronize. Delphi e C++Builder, Win32 e Win64.

Veja também

- Início rápido VCL
- Atualizações thread-safe
- Action links

TsgcVCLToolbar

Unit: sgcToolbar_VCL
TsgcVCLToolbar é a barra de ferramentas VCL, somente para Windows. Por padrão ela é alinhada ao topo e mostra seus Items da esquerda para a direita: consulte Tipos de item. Ela trata o mouse, o teclado, as dicas dos itens, os menus suspensos e o menu de excedentes. Sua altura segue ButtonSize e Padding, multiplicados por PPI / 96.

Propriedades

Propriedade	Tipo	Descrição
Items	TsgcToolbarItems	Os itens, na ordem de exibição. Consulte TsgcToolbarItems.
ShowCaptions	Boolean	Desenha o texto ao lado do ícone. Quando False, os botões com ícone mostram apenas o ícone e o texto passa a ser a dica. Um item sem Glyph e sem ImageIndex sempre mostra o seu texto. O padrão é True.
ButtonSize	Integer	Altura de um botão, e largura de um botão sem texto, em pixels lógicos. O padrão é 32.
Spacing	Integer	Espaço entre dois itens em pixels lógicos. O padrão é 4.
Padding	Integer	Espaço ao redor dos botões em pixels lógicos. A altura da barra é ButtonSize mais duas vezes Padding. O padrão é 6.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom, que usa as cores claras. Consulte Temas.
AccentColor	TsgcUIColor	Cor do anel de foco e de um alternador pressionado, no formato $AARRGGBB. 0 (padrão) usa o destaque do tema.
Images	TCustomImageList	A lista de imagens do ImageIndex do item. Cada imagem é convertida uma vez para ARGB direto e mantida em cache até que a lista mude. Consulte Imagens e glifos.
Model	TsgcToolbarModel (public, somente leitura)	O modelo da barra. Consulte TsgcToolbarModel.
OverflowMenu	TPopupMenu (public, somente leitura)	O menu do botão de excedentes, reconstruído toda vez que é exibido. Não está na edição Community. Consulte Excedentes.
MaxFPS	Integer	O limite de controle de quadros da base de controles do sgcUICore.

Os membros usuais de um controle também são publicados: **Align** (padrão alTop), **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **TabOrder**, **TabStop** (padrão False), **Visible**, OnClick, OnContextPopup, OnEnter, OnExit, OnMouseDown, OnMouseMove, OnMouseUp e OnResize.

Métodos

Método	Retorna	Descrição
ItemAtPos(X, Y: Integer)	Integer	O índice do item em um ponto do controle, CS_TOOLBAR_HIT_OVERFLOW (-2) para o botão de excedentes ou CS_TOOLBAR_HIT_NONE (-1). Separadores e espaçadores nunca são atingidos.
ItemRect(aIndex: Integer)	TRect	O retângulo de um item em coordenadas do controle, ou um retângulo vazio quando o item não está na barra.
OverflowRect	TRect	O retângulo do botão de excedentes, vazio quando não há excedentes.
ClickItem(aIndex: Integer)		Ativa um item como um clique faria: alterna um alternador, dispara OnItemClick, executa o OnClick do item ou sua Action e exibe o menu de um item suspenso.
InitiateAction		Faz com que cada item com uma Action se atualize. O framework a chama quando ocioso.

Eventos

Evento	Tipo	Descrição
OnItemClick	TsgcToolbarItemEvent	Um botão, alternador ou item suspenso habilitado foi clicado, tocado, escolhido no menu de excedentes ou ativado com Enter ou Espaço. O Down de um alternador já foi alterado. O OnClick do item ou sua Action é executado depois deste evento.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcFMXToolbar
- TsgcToolbarItem
- TsgcToolbarModel

- Início rápido VCL

TsgcFMXToolbar

Unit: sgcToolbar_FMX (Delphi XE6 ou posterior)
TsgcFMXToolbar é a barra de ferramentas FireMonkey, com as mesmas propriedades, métodos e eventos de TsgcVCLToolbar. Por padrão ela é alinhada ao topo e sua altura é ButtonSize mais duas vezes Padding, em unidades lógicas. Ela trata o mouse e o toque, o teclado quando pode receber foco, e exibe o TPopupMenu do FireMonkey de um item suspenso e do botão de excedentes.

Propriedades

Propriedade	Tipo	Descrição
Items	TsgcToolbarItems	Os itens, na ordem de exibição. Consulte TsgcToolbarItems.
ShowCaptions	Boolean	Desenha o texto ao lado do ícone. Quando False, os botões com ícone mostram apenas o ícone e o texto passa a ser a dica. Um item sem Glyph e sem ImageIndex sempre mostra o seu texto. O padrão é True.
ButtonSize	Integer	Altura de um botão, e largura de um botão sem texto, em pixels lógicos. O padrão é 32.
Spacing	Integer	Espaço entre dois itens em pixels lógicos. O padrão é 4.
Padding	Integer	Espaço ao redor dos botões em pixels lógicos. A altura da barra é ButtonSize mais duas vezes Padding. O padrão é 6.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom, que usa as cores claras. Consulte Temas.
AccentColor	TsgcUIColor	Cor do anel de foco e de um alternador pressionado, no formato $AARRGGBB. 0 (padrão) usa o destaque do tema.
Images	TCustomImageList	A lista de imagens do ImageIndex do item. Requer o XE8 ou posterior, e as imagens são desenhadas com 16 x 16 pixels lógicos. Consulte Imagens e glifos.
Model	TsgcToolbarModel (public, somente leitura)	O modelo da barra. Consulte TsgcToolbarModel.
OverflowMenu	TPopupMenu (public, somente leitura)	O menu do botão de excedentes, reconstruído toda vez que é exibido. Não está na edição Community. Consulte Excedentes.
MaxFPS	Integer	O limite de controle de quadros da base de controles do sgcUICore.

Os membros usuais de um controle FireMonkey também são publicados: **Align** (padrão Top), **Anchors**, **CanFocus** (padrão False), **Cursor**, **Enabled**, **HitTest**, **Locked**, **Margins**, **Opacity**, **PopupMenu**, **Position**, **TabOrder**, **Visible**, **Width** e, a partir do 10.1 Berlin, **TabStop**, **ParentShowHint** e **ShowHint**. Os eventos incluem OnClick, OnEnter, OnExit, OnKeyDown, OnKeyUp, os eventos de mouse, OnPainting, OnPaint e OnResize.

Métodos

Método	Retorna	Descrição
ItemAtPos(X, Y: Single)	Integer	O índice do item em um ponto do controle, CS_TOOLBAR_HIT_OVERFLOW (-2) para o botão de excedentes ou CS_TOOLBAR_HIT_NONE (-1). Separadores e espaçadores nunca são atingidos.
ItemRect(aIndex: Integer)	TRectF	O retângulo de um item em coordenadas do controle, ou um retângulo vazio quando o item não está na barra.
OverflowRect	TRectF	O retângulo do botão de excedentes, vazio quando não há excedentes.
ClickItem(aIndex: Integer)		Ativa um item como um clique faria: alterna um alternador, dispara OnItemClick, executa o OnClick do item ou sua Action e exibe o menu de um item suspenso.
InitiateAction		Faz com que cada item com uma Action se atualize. O framework a chama quando ocioso.

Eventos

Evento	Tipo	Descrição
OnItemClick	TsgcToolbarItemEvent	Um botão, alternador ou item suspenso habilitado foi clicado, tocado, escolhido no menu de excedentes ou ativado com Enter ou Espaço. O Down de um alternador já foi alterado. O OnClick do item ou sua Action é executado depois deste evento.

Exemplo

```
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;
```

Veja também

- TsgcVCLToolbar
- Início rápido FMX
- Plataformas

TsgcVCLStatusBar

Unit: sgcToolbar_VCL
TsgcVCLStatusBar é a barra de status VCL, somente para Windows. Por padrão ela é alinhada à base e mostra seus Panels: painéis de texto, painéis de progresso e painéis indicadores, fixos, AutoSize ou Spring. Ela mostra o Hint do painel sob o mouse e sua alça de redimensionamento redimensiona o formulário. Qualquer thread pode atualizar os painéis com os três setters thread-safe.

Propriedades

Propriedade	Tipo	Descrição
Panels	TsgcStatusBarPanels	Os painéis, da esquerda para a direita. Consulte TsgcStatusBarPanels.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom, que usa as cores claras. As cores dos indicadores seguem o tema.
AccentColor	TsgcUIColor	Cor das barras de progresso, no formato $AARRGGBB. 0 (padrão) usa o destaque do tema.
SizeGrip	Boolean	Desenha a alça de redimensionamento na extremidade direita. Ela redimensiona o formulário pai quando o formulário tem uma borda redimensionável e não está maximizado. O padrão é True. Consulte Alça de redimensionamento.
Height	Integer	Altura da barra, 24 pixels lógicos por padrão.
Model	TsgcStatusBarModel (public, somente leitura)	O modelo da barra. Consulte TsgcStatusBarModel.
MaxFPS	Integer	O limite de controle de quadros da base de controles do sgcUICore.

Os membros usuais de um controle também são publicados: **Align** (padrão alBottom), **Anchors**, **Constraints**, **Enabled**, **ParentShowHint**, **PopupMenu**, **ShowHint**, **Visible**, OnClick, OnContextPopup, OnMouseDown, OnMouseMove, OnMouseUp e OnResize.

Métodos

Método	Retorna	Descrição
SetPanelText(aIndex: Integer; const aText: string)		Define o Text de um painel. É thread-safe e um índice inválido é ignorado. Consulte Atualizações thread-safe.
SetProgress(aIndex, aValue: Integer)		Define o Progress (0 a 100) de um painel. É thread-safe.
SetIndicator(aIndex: Integer; aState: TsgcStatusIndicatorState)		Define o IndicatorState de um painel. É thread-safe.
PanelAtPos(X, Y: Integer)	Integer	O índice do painel em um ponto do controle, ou -1.

PanelRect(aIndex: Integer)	TRect	O retângulo de um painel em coordenadas do controle.
GripRect	TRect	O retângulo da alça de redimensionamento, vazio quando SizeGrip é False.

Eventos

Evento	Tipo	Descrição
OnPanelClick	TsgcStatusBarPanelEvent	Um painel foi clicado ou tocado. aPanel é o painel.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcFMXStatusBar
- TsgcStatusBarPanel
- Atualizações thread-safe
- TsgcStatusBarModel

TsgcFMXStatusBar

Unit: sgcToolbar_FMX (Delphi XE6 ou posterior)
TsgcFMXStatusBar é a barra de status FireMonkey, com as mesmas propriedades, métodos e eventos de TsgcVCLStatusBar. Por padrão ela é alinhada à base. Os setters thread-safe enfileiram a repintura para a thread principal. A alça de redimensionamento é desenhada, mas não redimensiona o formulário.

Propriedades

Propriedade	Tipo	Descrição
Panels	TsgcStatusBarPanels	Os painéis, da esquerda para a direita. Consulte TsgcStatusBarPanels.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom, que usa as cores claras. As cores dos indicadores seguem o tema.
AccentColor	TsgcUIColor	Cor das barras de progresso, no formato $AARRGGBB. 0 (padrão) usa o destaque do tema.
SizeGrip	Boolean	Desenha a alça de redimensionamento na extremidade direita. O FireMonkey não tem teste de posição na área não cliente, portanto a alça não redimensiona o formulário. O padrão é True. Consulte Alça de redimensionamento.
Height	Single	Altura da barra, 24 pixels lógicos por padrão.
Model	TsgcStatusBarModel (public, somente leitura)	O modelo da barra. Consulte TsgcStatusBarModel.
MaxFPS	Integer	O limite de controle de quadros da base de controles do sgcUICore.

Os membros usuais de um controle FireMonkey também são publicados: **Align** (padrão Bottom), **Anchors**, **Cursor**, **Enabled**, **HitTest**, **Locked**, **Margins**, **Opacity**, **PopupMenu**, **Position**, **Visible**, **Width** e, a partir do 10.1 Berlin, **Size**, **ParentShowHint** e **ShowHint**. Os eventos incluem OnClick, os eventos de mouse, OnPainting, OnPaint e OnResize.

Métodos

Método	Retorna	Descrição
SetPanelText(aIndex: Integer; const aText: string)		Define o Text de um painel. É thread-safe e um índice inválido é ignorado. Consulte Atualizações thread-safe.
SetProgress(aIndex, aValue: Integer)		Define o Progress (0 a 100) de um painel. É thread-safe.
SetIndicator(aIndex: Integer; aState: TsgcStatusIndicatorState)		Define o IndicatorState de um painel. É thread-safe.
PanelAtPos(X, Y: Single)	Integer	O índice do painel em um ponto do controle, ou -1.

PanelRect(aIndex: Integer)	TRectF	O retângulo de um painel em coordenadas do controle.
GripRect	TRectF	O retângulo da alça de redimensionamento, vazio quando SizeGrip é False.

Eventos

Evento	Tipo	Descrição
OnPanelClick	TsgcStatusBarPanelEvent	Um painel foi clicado ou tocado. aPanel é o painel.

Exemplo

A parte da barra de status do início rápido do FireMonkey: um painel de texto elástico e um painel de progresso.

```
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;
```

Veja também

- TsgcVCLStatusBar
- Início rápido FMX
- Alça de redimensionamento

TsgcToolbarItem

Unit: sgcToolbar_Model
TsgcToolbarItem é um item de uma barra de ferramentas, um TCollectionItem de TsgcToolbarItems. A mesma classe é usada pelas barras de ferramentas VCL e FMX.

Propriedades

Propriedade	Tipo	Descrição
Kind	TsgcToolbarItemKind	tikButton (padrão), tikToggle, tikDropdown, tikSeparator, tikSpacer ou tikLabel. Consulte Tipos de item.
Caption	string	O texto do item.
Glyph	string	Um texto curto desenhado no espaço do ícone quando o item não tem imagem, por exemplo B para Negrito.
ImageIndex	Integer	Índice em Images da barra, -1 (padrão) para nenhum.
Hint	string	A dica do item. Quando está vazia e os textos estão ocultos, o Caption é a dica.
Enabled	Boolean	Um item desabilitado é desenhado esmaecido e ignora o mouse e o teclado. O padrão é True.
Visible	Boolean	Um item oculto não ocupa espaço. O padrão é True.
Down	Boolean	O estado de um item tikToggle. Com GroupIndex maior que 0, definir Down desmarca os outros itens do mesmo grupo. O padrão é False.
GroupIndex	Integer	Alternadores com o mesmo GroupIndex maior que 0 formam um grupo de rádio. O padrão é 0. Consulte Alternadores e grupos de rádio.
Width	Integer	Largura em pixels lógicos, 0 (padrão) para automática. Para um espaçador, é a largura mínima.
Tag	Integer	Um valor livre para o seu código.
DropdownMenu	TComponent	O menu popup de um item tikDropdown: um TPopupMenu da VCL em TsgcVCLToolbar, um TPopupMenu do FireMonkey em TsgcFMXToolbar. A barra limpa a referência quando o menu é liberado. Consulte Menus suspensos.
Action	TBasicAction	Uma ação vinculada. Caption, Enabled, Visible, Hint, ImageIndex e Checked (Down de um alternador) a seguem, e um clique a executa, a menos que OnClick esteja atribuído. Não está na edição Community. Consulte Action links.
BoundsRect	TsgcUIRect (public, somente leitura)	O retângulo do último layout, em pixels do dispositivo.

Shown	Boolean (public, somente leitura)	True quando o último layout colocou o item na barra.
InOverflow	Boolean (public, somente leitura)	True quando o último layout moveu o item para a lista de excedentes.

Métodos

Método	Retorna	Descrição
Click(aComponent: TComponent = nil)		Executa OnClick quando está atribuído; caso contrário, executa a Action vinculada com aComponent como seu ActionComponent. Ele não alterna um alternador nem dispara OnItemClick, use ClickItem da barra para isso.
InitiateAction		Faz com que a Action vinculada se atualize. Não faz nada sem uma Action.
IsButton	Boolean	True para os tipos que podem ser clicados: tikButton, tikToggle e tikDropdown.

Eventos

Evento	Tipo	Descrição
OnClick	TNotifyEvent	O item foi ativado. Sender é o item. É executado depois do OnItemClick da barra.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcToolbarItems
- Tipos de item
- Alternadores e grupos de rádio

TsgcToolbarItems

Unit: sgcToolbar_Model
TsgcToolbarItems é a coleção de TsgcToolbarItem de uma barra de ferramentas e de TsgcToolbarModel. Uma alteração em um item atualiza a barra.

Propriedades

Propriedade	Tipo	Descrição
Items[Index: Integer]	TsgcToolbarItem	O item em Index. É a propriedade padrão, portanto Toolbar.Items[0] é o primeiro item.
Count	Integer	O número de itens.

Métodos

Método	Retorna	Descrição
Add	TsgcToolbarItem	Adiciona um item tikButton no final e o retorna.
Delete(Index: Integer)		Remove um item.
Clear		Remove todos os itens.

Para alterar muitos itens com uma única repintura, envolva as alterações em BeginUpdate e EndUpdate do modelo.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcToolbarItem
- TsgcToolbarModel

TsgcStatusBarPanel

Unit: sgcToolbar_Model
TsgcStatusBarPanel é um painel de uma barra de status, um TCollectionItem de TsgcStatusBarPanels. Text, Progress e IndicatorState são protegidos pelo lock do modelo.

Propriedades

Propriedade	Tipo	Descrição
Kind	TsgcStatusPanelKind	spkText (padrão), spkProgress ou spkIndicator. Consulte Progresso e indicadores.
Text	string	O texto de um painel de texto ou indicador.
Width	Integer	Largura em pixels lógicos quando nem AutoSize nem Spring estão definidos. O padrão é 100.
AutoSize	Boolean	A largura acompanha o texto e o ponto de um indicador. O padrão é False.
Spring	Boolean	Os painéis elásticos dividem a largura que sobra dos outros. O padrão é False. Consulte Painéis de status.
Alignment	TAlignment	taLeftJustify (padrão), taCenter ou taRightJustify.
Progress	Integer	O valor de um painel spkProgress, de 0 a 100. O padrão é 0.
IndicatorState	TsgcStatusIndicatorState	O estado de um painel spkIndicator: sisOff (padrão), sisOk, sisWarning ou sisError.
Hint	string	A dica exibida sobre o painel.
Tag	Integer	Um valor livre para o seu código.
BoundsRect	TsgcUIRect (public, somente leitura)	O retângulo do último layout, em pixels do dispositivo.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcStatusBarPanels
- Painéis de status
- Progresso e indicadores

TsgcStatusBarPanels

Unit: sgcToolbar_Model
TsgcStatusBarPanels é a coleção de TsgcStatusBarPanel de uma barra de status e de TsgcStatusBarModel.

Propriedades

Propriedade	Tipo	Descrição
Items[Index: Integer]	TsgcStatusBarPanel	O painel em Index. É a propriedade padrão, portanto StatusBar.Panels[0] é o primeiro painel.
Count	Integer	O número de painéis.

Métodos

Método	Retorna	Descrição
Add	TsgcStatusBarPanel	Adiciona um painel spkText no final e o retorna.
Delete(Index: Integer)		Remove um painel.
Clear		Remove todos os painéis.

Adicione, exclua e configure painéis a partir da thread principal. A partir de uma thread de trabalho, use SetPanelText, SetProgress e SetIndicator da barra, consulte Atualizações thread-safe.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcStatusBarPanel
- TsgcStatusBarModel

TsgcToolbarModel

Unit: sgcToolbar_Model
TsgcToolbarModel é o modelo independente de framework de uma barra de ferramentas. Cada barra de ferramentas possui um, disponível pela sua propriedade **Model**, e um modelo criado em código desenha uma barra de ferramentas sem janela. Os tamanhos são pixels lógicos. Layout os multiplica pela escala, e todo retângulo calculado está em pixels do dispositivo.

Propriedades

Propriedade	Tipo	Descrição
Items, ShowCaptions, ButtonSize, Spacing, Padding, Theme, AccentColor		As propriedades publicadas da barra de ferramentas, consulte TsgcVCLToolbar.
HotIndex, PressedIndex, FocusedIndex	Integer	O estado de interação: um índice de item, CS_TOOLBAR_HIT_OVERFLOW para o botão de excedentes, ou -1.
HasOverflow	Boolean (somente leitura)	True quando o último layout moveu itens para a lista de excedentes.
OverflowCount	Integer (somente leitura)	O número de itens na lista de excedentes.
OverflowItem[Index: Integer]	TsgcToolbarItem (somente leitura)	Um item da lista de excedentes.
OverflowRect	TsgcUIRect (somente leitura)	O botão de excedentes do último layout, em pixels do dispositivo.
Owner	TPersistent	O componente que possui o modelo, usado para o caminho de nomes em tempo de design.

Métodos

Método	Retorna	Descrição
Layout(aWidth, aHeight: Integer; aScale: Double; aMeasure or aCanvas)		Calcula os retângulos dos itens para uma barra de aWidth por aHeight pixels do dispositivo. aMeasure (um TsgcToolbarMeasureEvent) ou aCanvas mede o texto; um measure nil usa uma estimativa de 0,55 vez o tamanho da fonte por caractere. Os espaçadores dividem a largura livre. Com o menu de excedentes, os itens que não cabem vão para a lista de excedentes; caso contrário, são cortados.

HitTest(aX, aY: Double)	Integer	O item em um ponto do último layout, CS_TOOLBAR_HIT_OVERFLOW ou CS_TOOLBAR_HIT_NONE.
Render(aCanvas; aWidth, aHeight: Integer; aScale: Double = 1)		Desenha a barra em um canvas do sgcUICore, usando OnGetImage para as imagens.
SaveToPNG, SaveToSVG(aFileName or aStream; aWidth, aHeight: Integer; aScale: Double = 1)		Grava a barra como uma imagem. Consulte Renderização headless.
BeginUpdate, EndUpdate		Agrupa as alterações em um único OnChange.
GetAccent	TsgcUIColor	AccentColor, ou o destaque do tema quando é 0.
ShowsCaption(aItem)	Boolean	True quando o item desenha o seu texto: ShowCaptions, ou o item não tem ícone.
HasIcon(aItem)	Boolean	True quando o item tem um Glyph ou um ImageIndex.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O modelo mudou. As barras se repintam com ele.
OnGetImage	TsgcToolbarImageEvent	Retorna a imagem de um ImageIndex como um TsgcUIImage para Render e para a saída headless.

Exemplo

Layout e HitTest sem um controle, como fazem um teste ou um 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;
```

Veja também

- TsgcVCLToolbar
- Renderização headless
- Arquitetura

TsgcStatusBarModel

Unit: sgcToolbar_Model
TsgcStatusBarModel é o modelo independente de framework de uma barra de status. Cada barra de status possui um, disponível pela sua propriedade **Model**. Seus três setters são thread-safe: eles obtêm o lock do modelo, alteram o valor e disparam OnChange a partir da thread chamadora, e somente quando o valor mudou. Todas as outras propriedades destinam-se à thread principal.

Propriedades

Propriedade	Tipo	Descrição
Panels, Theme, AccentColor, SizeGrip		As propriedades publicadas da barra de status, consulte TsgcVCLStatusBar.
GripRect	TsgcUIRect (somente leitura)	A alça de redimensionamento do último layout em pixels do dispositivo, vazia quando SizeGrip é False.
Owner	TPersistent	O componente que possui o modelo, usado para o caminho de nomes em tempo de design.

Métodos

Método	Retorna	Descrição
SetPanelText, SetProgress, SetIndicator		Os setters thread-safe. Um índice inválido é ignorado.
Layout(aWidth, aHeight: Integer; aScale: Double; aMeasure or aCanvas)		Calcula os retângulos dos painéis: primeiro as larguras fixas, depois os painéis AutoSize conforme o texto, e então os painéis Spring dividem o que sobra, nunca abaixo de 0. A alça de redimensionamento é reservada à direita quando SizeGrip é True.
HitTest(aX, aY: Double)	Integer	O painel em um ponto do último layout, ou -1.
Render, SaveToPNG, SaveToSVG		Desenham a barra em um canvas ou a gravam como uma imagem, como em TsgcToolbarModel.
GetIndicatorColor(aState)	TsgcUIColor	A cor de um estado de indicador no tema atual.
GetAccent	TsgcUIColor	AccentColor, ou o destaque do tema quando é 0.
BeginUpdate, EndUpdate		Agrupa as alterações em um único OnChange.
Lock, Unlock		O lock que protege Text, Progress e IndicatorState.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O modelo mudou. Depois de um setter thread-safe, ele é disparado na thread chamadora, e as barras encaminham a repintura para a thread principal.

Exemplo

```
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;
```

Veja também

- TsgcVCLStatusBar
- Atualizações thread-safe
- Renderização headless

Tipos de item

O **Kind** de um TsgcToolbarItem decide como ele é desenhado e como reage. Os itens são posicionados da esquerda para a direita na ordem da coleção, com Spacing pixels lógicos entre eles.

Kind	Comportamento
tikButton	O padrão. Um clique dispara o OnItemClick da barra e, em seguida, o OnClick do item ou a sua Action.
tikToggle	Um botão que permanece pressionado. Um clique alterna Down antes de OnItemClick ser disparado. Com GroupIndex maior que 0, ele é um botão de rádio, consulte Alternadores e grupos de rádio.
tikDropdown	Um botão com uma pequena seta que mostra o seu DropdownMenu abaixo do botão. Consulte Menus suspensos.
tikSeparator	Uma linha vertical fina. Ele nunca é clicado.
tikSpacer	Um item vazio que ocupa a largura livre da barra, dividida com os outros espaçadores. Ele empurra os itens seguintes para a direita. Width é a sua largura mínima.
tikLabel	Um texto sem botão, por exemplo antes de um campo de edição. Ele nunca é clicado.

Um botão desenha o seu ícone (a imagem de ImageIndex ou o texto do Glyph) e o seu Caption. Com ShowCaptions False, apenas o ícone é desenhado e o texto passa a ser a dica. Um item sem ícone sempre desenha o seu texto. Width maior que 0 fixa a largura de um item em pixels lógicos.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcToolbarItem
- Imagens e glifos
- Excedentes

Alternadores e grupos de rádio

Um item **tikToggle** tem um estado **Down**, desenhado com um tom de destaque. Um clique alterna Down e depois dispara OnItemClick, de modo que o manipulador lê o novo estado.
Alternadores que compartilham um **GroupIndex** maior que 0 formam um grupo de rádio: definir Down em um deles, por um clique ou por código, desmarca os outros. Um clique em um alternador de rádio que já está pressionado o mantém pressionado, de modo que um item do grupo permanece selecionado. Um alternador vinculado a uma Action segue o Checked da ação.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- Tipos de item
- Action links
- TsgcToolbarItem

Menus suspensos

Um item **tikDropdown** desenha uma pequena seta após o seu texto e mostra o menu popup da sua propriedade **DropdownMenu** abaixo do botão: um TPopupMenu da VCL em TsgcVCLToolbar, um TPopupMenu do FireMonkey em TsgcFMXToolbar. A propriedade é declarada como TComponent para que o item permaneça independente de framework. A barra limpa a referência quando o menu é liberado.
No VCL, o menu abre quando o botão é pressionado. OnItemClick e o OnClick do item são disparados primeiro, de modo que um manipulador pode preencher ou marcar o menu antes de ele aparecer. Um item suspenso na lista de excedentes mostra o seu menu abaixo do botão de excedentes.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- Tipos de item
- Excedentes
- Solução de problemas

Excedentes

Quando os itens não cabem na largura da barra, o layout move os últimos para a lista de excedentes e reserva um botão de excedentes, desenhado com um chevron duplo, na extremidade direita. Pressioná-lo mostra o **OverflowMenu** da barra, um menu popup construído em tempo de execução com uma entrada para cada item da lista. Escolher uma entrada ativa aquele item exatamente como um clique: um alternador é alternado, OnItemClick é disparado e um item suspenso mostra o seu menu.
A lista de excedentes do último layout está disponível pelo modelo: **HasOverflow**, **OverflowCount**, **OverflowItem**[i] e a propriedade **InOverflow** de cada item. ItemAtPos retorna CS_TOOLBAR_HIT_OVERFLOW sobre o botão de excedentes.

Edição Community

O menu de excedentes não faz parte da edição Community (SGC_TOOLBAR_ADVANCED). Nela, os itens que não cabem são cortados na borda direita e a propriedade OverflowMenu não existe. Consulte Edições.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcToolbarModel
- Menus suspensos
- Edições

Action links

A propriedade **Action** vincula um item a qualquer TBasicAction, um TAction da VCL ou do FireMonkey, sem que o modelo dependa de nenhum dos dois frameworks. O item passa então a seguir **Caption**, **Enabled**, **Visible**, **Hint**, **ImageIndex** e **Checked** da ação, em que Checked controla o Down de um item tikToggle. Um TAction do FireMonkey fornece o seu Text como Caption.
Um clique executa a ação, com a barra como seu ActionComponent, a menos que o item tenha um manipulador OnClick. As barras chamam InitiateAction quando ociosas, de modo que o OnUpdate de cada ação vinculada é executado e uma ação desabilitada desabilita o seu botão. A partir do Delphi XE3, a ação envia suas alterações ao item. Antes do XE3, o item copia as propriedades da ação quando ocioso.

Edição Community

Os Action links não fazem parte da edição Community (SGC_TOOLBAR_ADVANCED): a propriedade Action não existe. Use o OnClick do item ou o OnItemClick da barra. Consulte Edições.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcToolbarItem
- Alternadores e grupos de rádio
- Solução de problemas

Imagens e glifos

Todo item clicável tem um espaço de ícone de 16 pixels lógicos. Ele mostra a imagem em **ImageIndex** da lista **Images** da barra ou, quando o item não tem imagem, o seu **Glyph**: um texto curto, como uma letra ou um símbolo. Um item sem nenhum dos dois mostra apenas o seu texto.

- No VCL, Images é um TCustomImageList. Cada imagem é convertida uma vez para ARGB direto, com o alfa recuperado, e mantida em cache até que a lista de imagens mude.
- No FMX, Images é o TCustomImageList do FireMonkey de FMX.ImgList e requer o Delphi XE8 ou posterior. As imagens são obtidas no tamanho do dispositivo do quadro. Em versões mais antigas, os itens desenham o seu Glyph.
- A saída headless desenha o Glyph, a menos que o evento OnGetImage do modelo retorne uma imagem.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- Tipos de item
- Renderização headless
- Plataformas

Painéis de status

Uma barra de status organiza seus Panels da esquerda para a direita em três passagens:

- Os painéis **fixos**, sem AutoSize nem Spring, ocupam a sua **Width** em pixels lógicos (100 por padrão).
- Os painéis **AutoSize** ocupam a largura do seu texto, mais o ponto de um indicador.
- Os painéis **Spring** dividem a largura que sobra, nunca abaixo de 0. Use um painel Spring para a mensagem principal.

Uma linha fina separa os painéis. O texto de um painel é alinhado com **Alignment**, e **Hint** é exibida quando o mouse fica sobre o painel e ShowHint é True. OnPanelClick informa um clique em um painel. Quando SizeGrip é True, a alça é reservada na extremidade direita, consulte Alça de redimensionamento.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcStatusBarPanel
- Progresso e indicadores
- Atualizações thread-safe

Progresso e indicadores

Além dos painéis de texto (spkText), um painel pode ser uma barra de progresso ou um indicador de estado.

Kind	Desenha
spkProgress	Uma barra preenchida até **Progress** por cento (0 a 100), na AccentColor da barra de status ou no destaque do tema.
spkIndicator	Um ponto colorido seguido do **Text** do painel. A cor vem de **IndicatorState**, que é adequado ao estado de uma conexão ou de um feed de dados.

IndicatorState	Tema claro	Tema escuro
sisOff	A cor de texto esmaecida do tema	A cor de texto esmaecida do tema
sisOk	Verde $FF1A7F37	Verde $FF3FB950
sisWarning	Âmbar $FFBF8700	Âmbar $FFD29922
sisError	Vermelho $FFCF222E	Vermelho $FFF85149

GetIndicatorColor de TsgcStatusBarModel retorna a cor de um estado no tema atual.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- Painéis de status
- Atualizações thread-safe
- Temas

Atualizações thread-safe

Uma aplicação em tempo real atualiza a sua barra de status a partir de threads de trabalho: um download, um feed, um cálculo demorado. **SetPanelText**, **SetProgress** e **SetIndicator** da barra de status podem ser chamados de qualquer thread, sem Synchronize nem Queue.

- O modelo protege Text, Progress e IndicatorState dos painéis com um lock.
- O setter dispara o OnChange do modelo a partir da thread chamadora, e somente quando o valor mudou, portanto repetir o mesmo valor não custa nada.
- A barra responde com uma solicitação de repintura. A partir de uma thread de trabalho, a barra VCL apenas posta uma mensagem à sua janela e a barra FMX enfileira a repintura para a thread principal, de modo que o framework nunca é acessado fora da thread principal.
- Um índice fora do intervalo é ignorado, portanto uma thread que sobrevive a um painel não gera exceção.

Todas as outras propriedades, assim como adicionar, excluir ou configurar painéis, devem ser feitas a partir da thread principal. Certifique-se de que a thread de trabalho pare antes que o formulário que possui a barra de status seja destruído.

Exemplo

Uma thread de trabalho que simula um download de 50 blocos, como no demo LiveStatus. O painel 0 é um painel de texto elástico, o painel 1 um painel de progresso e o painel 2 um 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;
```

Veja também

- TsgcStatusBarModel
- Progresso e indicadores
- Demos

Alça de redimensionamento

Com **SizeGrip** True, o padrão, a barra de status reserva uma pequena área na extremidade direita e desenha ali os pontos da alça. **GripRect** retorna o seu retângulo, vazio quando SizeGrip é False.

- No VCL, TsgcVCLStatusBar responde ao teste de posição na área não cliente da alça, de modo que arrastá-la redimensiona o formulário pai. Isso só ocorre quando o formulário tem uma borda redimensionável e não está maximizado, e nunca em tempo de design.
- No FMX, TsgcFMXStatusBar apenas desenha a alça. O FireMonkey não tem teste de posição na área não cliente, portanto a alça não redimensiona o formulário. Defina SizeGrip como False quando o formulário não for redimensionável.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- Painéis de status
- TsgcVCLStatusBar

Teclado

A barra de ferramentas é alcançada com a tecla Tab quando **TabStop** é True no VCL, ou **CanFocus** é True no FMX. Os dois são False por padrão, de modo que a barra de ferramentas não tira o foco do editor abaixo dela. Quando a barra de ferramentas tem o foco, o botão focado tem um anel de foco:

Tecla	Ação
Left, Right	Move para o botão, alternador ou item suspenso habilitado anterior ou seguinte.
Home, End	Move para o primeiro ou para o último.
Enter, Space	Ativa o item focado, como um clique faria.

ClickItem ativa um item por código, por exemplo a partir de um atalho de teclado do formulário. A barra de status não tem interface de teclado.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

- TsgcVCLToolbar
- TsgcFMXToolbar

Temas

As barras obtêm suas cores dos tokens de tema do sgcUICore, os mesmos usados pelos outros produtos da suíte. **Theme** é utLight (padrão), utDark ou utCustom, que usa as cores claras. Cada barra tem o seu próprio Theme, portanto defina-o na barra de ferramentas e na barra de status.

- O fundo é a cor Surface, com uma linha Border abaixo da barra de ferramentas e acima da barra de status.
- Os botões sob o mouse e pressionados recebem uma sobreposição, um alternador pressionado recebe um tom de destaque, o botão focado recebe um anel de foco e um item desabilitado usa a cor de texto esmaecida.
- **AccentColor**, no formato $AARRGGBB, substitui o destaque do tema: o anel de foco e o tom de pressionado na barra de ferramentas, as barras de progresso na barra de status. 0 usa o destaque do tema.
- As cores dos indicadores seguem o tema, consulte Progresso e indicadores.

Exemplo

O exemplo é um método de um formulário VCL com uma **Toolbar** (TsgcVCLToolbar), uma **StatusBar** (TsgcVCLStatusBar) e um **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;
```

Veja também

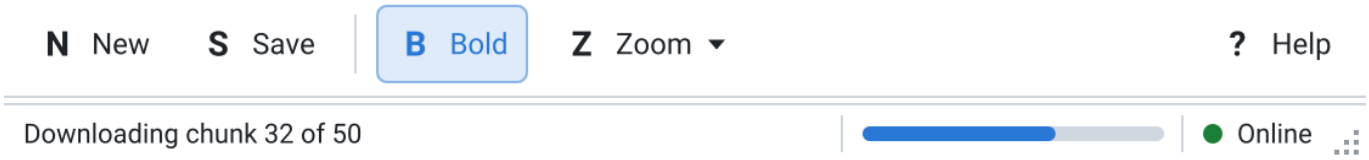
- Progresso e indicadores
- Arquitetura

Renderização headless

O renderizador em sgcToolbar_Render desenha um TsgcToolbarModel ou um TsgcStatusBarModel sem janela e sem VCL ou FMX, de modo que uma aplicação de console, um serviço, um teste ou um servidor web pode produzir as mesmas barras como imagem, por exemplo para a documentação de uma aplicação ou para um teste visual. As units compilam em todas as plataformas, inclusive no Linux64.

Rotina	Descrição
Model.SaveToPNG(aFileName or aStream, aWidth, aHeight, aScale = 1)	Grava um arquivo ou stream PNG. aWidth e aHeight são pixels lógicos e aScale os multiplica: 560 por 44 na escala 2 resulta em uma imagem de 1120 por 88.
Model.SaveToSVG(aFileName or aStream, aWidth, aHeight, aScale = 1)	Grava um arquivo ou stream SVG da mesma barra.
Model.Render(aCanvas, aWidth, aHeight, aScale = 1)	Desenha a barra em qualquer canvas do sgcUICore, em pixels do dispositivo.
RenderToolbar(aModel, aCanvas, aWidth, aHeight, aScale, aImages)	O renderizador de sgcToolbar_Render. aImages retorna a imagem de um ImageIndex, nil desenha o Glyph.
RenderStatusBar(aModel, aCanvas, aWidth, aHeight, aScale)	O renderizador da barra de status.

Uma barra de ferramentas tem altura ButtonSize mais duas vezes Padding, 44 pixels lógicos com os valores padrão. O modelo de um controle está disponível pela sua propriedade **Model**, portanto **Toolbar.Model.SaveToPNG(...)** salva o que o controle mostra em qualquer tamanho.



Exemplo

```
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;
```

Veja também

- TsgcToolbarModel
- TsgcStatusBarModel
- Plataformas

Visão geral do .NET

O **sgcToolbar .NET** é um port gerenciado em C# da biblioteca sgcToolbar do Delphi, e não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcToolbar** e mantêm os nomes do Delphi, então **TsgcToolbarModel**, **TsgcToolbarItem**, **TsgcToolbarItemKind**, **TsgcStatusBarModel** e **TsgcStatusBarPanel** têm a mesma aparência nas duas linguagens, e a barra de ferramentas e a barra de status de cada host seguem o padrão de nomes do Delphi. Elas são desenhadas com o port .NET do **sgcUICore**, com o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe, de modo que uma barra tem a mesma aparência no WinForms, no WPF, no Avalonia e na saída headless em PNG e SVG.
Uma barra de ferramentas contém botões, alternadores, grupos de rádio, botões suspensos, separadores, espaçadores e rótulos, e move os itens que não cabem para um menu de excedentes. Uma barra de status contém painéis de texto, de progresso e de indicador com largura fixa, automática ou elástica, e seus painéis podem ser atualizados a partir de qualquer thread.

Pacotes

O port está dividido em cinco pacotes NuGet, de modo que um projeto só traz o host que utiliza.

PackageId	Conteúdo	Frameworks de destino
esegece.sgcToolbar	Os modelos (**TsgcToolbarModel**, **TsgcStatusBarModel**), seus itens e painéis, o vínculo **Command** do item, o renderizador **sgcToolbar_Render** e a saída SVG headless. Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcToolbar.Rendering	A saída PNG headless, os métodos de extensão **SaveToPNG** de **sgcToolbar_Raster**, desenhados no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, no Linux e no macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcToolbar.WinForms	**TsgcWinFormsToolbar** e **TsgcWinFormsStatusBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcToolbar.Wpf	**TsgcWpfToolbar** e **TsgcWpfStatusBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcToolbar.Avalonia	**TsgcAvaloniaToolbar** e **TsgcAvaloniaStatusBar**, no Windows, no Linux e no macOS com o Avalonia 11.	net8.0, net9.0

Os três pacotes de host dependem de esegece.sgcToolbar.Rendering, que depende de esegece.sgcToolbar, de modo que uma única linha `PackageReference` traz tudo o que um host precisa. Veja Instalação pelo NuGet.

Edições

Edição	Ids dos pacotes	Inclui

Registered	esegece.sgcToolbar, esegece.sgcToolbar.Rendering, esegece.sgcToolbar.WinForms, esegece.sgcToolbar.Wpf, esegece.sgcToolbar.Avalonia	Tudo: as barras de ferramentas e de status dos três hosts, todos os tipos de item e de painel, grupos de rádio, menus suspensos, o menu de excedentes, o vínculo Command do item, imagens, dicas, navegação pelo teclado, os setters thread-safe da barra de status, a alça de redimensionamento, os temas e a saída headless em PNG e SVG.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcToolbar.WinForms.Community. Eles dependem dos pacotes Community do sgcUICore.	O mesmo conteúdo da Registered, sem o menu de excedentes e sem as propriedades **Command** e **CommandParameter** do item, como na edição Community do Delphi. Os itens que não cabem são cortados na borda direita, e os itens executam o seu OnClick.

A edição Community é gratuita para uso comercial e não comercial por indivíduos e organizações com menos de EUR 1.000.000 de receita bruta anual e no máximo cinco desenvolvedores. Uma organização que atinja ou ultrapasse qualquer um dos limites precisa de uma licença Registered. O port .NET não tem edição Trial.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	VCL (TsgcVCLToolbar, TsgcVCLStatusBar)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	FMX (TsgcFMXToolbar, TsgcFMXStatusBar)	Windows, .NET Framework 4.7.2 e .NET 8 e 9. As barras são declaradas em XAML e os itens e painéis são adicionados em código.
Avalonia	FMX (TsgcFMXToolbar, TsgcFMXStatusBar)	Windows, Linux e macOS, .NET 8 e 9, com mouse, toque e caneta. As barras são declaradas em XAML e os itens e painéis são adicionados em código.
Headless	Os métodos Render e SaveToSVG dos modelos	Qualquer plataforma dos frameworks de destino: SVG com esegece.sgcToolbar, PNG com esegece.sgcToolbar.Rendering.

Toda barra de ferramentas tem a mesma API: Items, ShowCaptions, ButtonSize, Spacing, Padding, Theme, AccentColor, Images, OverflowMenu, ItemAtPos, ItemRect, OverflowRect, ClickItem e OnItemClick. Toda barra de status tem Panels, SizeGrip, Theme, AccentColor, PanelAtPos, PanelRect, GripRect, os métodos thread-safe SetPanelText, SetProgress e SetIndicator, e OnPanelClick. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Renderização headless no .NET
- Demos do .NET
- Plataformas

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, e os pacotes core e de renderização vêm junto como dependências.

Edição Registered

```
dotnet add package esegece.sgcToolbar.WinForms
dotnet add package esegece.sgcToolbar.Wpf
dotnet add package esegece.sgcToolbar.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcToolbar.WinForms.Community
dotnet add package esegece.sgcToolbar.Wpf.Community
dotnet add package esegece.sgcToolbar.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcToolbar.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie em um projeto os pacotes Registered ou os Community, nunca os dois: eles contêm os mesmos tipos no mesmo namespace. O código que usa o **Command** do item ou a propriedade **OverflowMenu** de uma barra de ferramentas não compila com os pacotes Community; proteja-o com `#if !SGC_COMMUNITY` quando um projeto compila as duas edições.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:sgc="clr-namespace:esegece.sgcToolbar;assembly=esegece.sgcToolbar.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicativo desktop Avalonia. Mapeie o namespace no XAML com `xmlns:sgc="using:esegece.sgcToolbar"`.
- Headless: esegece.sgcToolbar sozinho basta para SVG. Adicione esegece.sgcToolbar.Rendering para PNG, ele traz o SkiaSharp e seus recursos nativos para Windows e Linux.

Em C# todo host precisa de `using esegece.sgcToolbar;`, e de `using esegece.sgcUICore;` para **TsgcUITheme** e os demais tipos dos temas.

Veja também

- Visão geral do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Demos do .NET

Início rápido WinForms

TsgcWinFormsToolbar e **TsgcWinFormsStatusBar** são as barras do WinForms, os equivalentes de TsgcVCLToolbar e TsgcVCLStatusBar do Delphi. Cada uma possui um modelo, expõe suas propriedades e o desenha por meio do sgcUICore. O pacote é **esegece.sgcToolbar.WinForms**.

Passos

- Crie um projeto WinForms e adicione o pacote esegece.sgcToolbar.WinForms.
- Crie um **TsgcWinFormsToolbar**. Ele é ancorado no topo por padrão e sua altura segue **ButtonSize** e **Padding**.
- Adicione os itens com `Items.Add()` e defina seu **Kind**, **Caption**, **Glyph** ou **ImageIndex**, e **Hint**. Um **tikToggle** com o mesmo **GroupIndex** forma um grupo de rádio, e um **tikDropdown** mostra o **ContextMenuStrip** de seu **DropdownMenu**.
- Crie um **TsgcWinFormsStatusBar**, que é ancorado na parte inferior, e adicione seus painéis com `Panels.Add()`.
- Atribua **OnItemClick** para reagir a todos os cliques, ou **OnClick** de cada item.

Exemplo

O Program.cs completo de um editor WinForms com uma barra de ferramentas e uma barra de status: dois botões, um alternador Bold, um grupo de rádio Left e Center, um botão suspenso Zoom, e um painel de texto elástico, um painel de progresso e um indicador. Ele compila como está em um projeto `net8.0-windows` que referencia esegece.sgcToolbar.WinForms.
As barras também estão na caixa de ferramentas. Os itens e painéis não são serializados pelo designer, portanto são sempre adicionados em código, veja Diferenças do .NET em relação ao Delphi. Defina **TabStop** como True para mover o foco entre os botões com as teclas de seta.

Veja também

- Visão geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início rápido WPF
- Início rápido Avalonia
- Tipos de item

Início rápido WPF

TsgcWpfToolbar e **TsgcWpfStatusBar** são as barras do WPF. Elas são declaradas em XAML como qualquer controle, e seus itens e painéis são adicionados em código. O pacote é **esegece.sgcToolbar.Wpf**.

Passos

- Crie um projeto WPF (`dotnet new wpf`) e adicione o pacote esegece.sgcToolbar.Wpf.
- Mapeie o namespace `esegece.sgcToolbar` e declare um **TsgcWpfToolbar** e um **TsgcWpfStatusBar**. **ShowCaptions**, **ButtonSize**, **SizeGrip** e as outras propriedades simples podem ser definidas como atributos XAML.
- No code-behind atribua **Images**, uma lista de **ImageSource**, adicione os itens e os painéis e atribua **OnItemClick**: é um delegate, não um evento do WPF, portanto não pode ser definido como atributo XAML.

Exemplo

MainWindow.xaml declara a barra de ferramentas, a barra de status e uma caixa de texto. O App.xaml do projeto é o do template WPF, com StartupUri definido como MainWindow.xaml.
MainWindow.xaml.cs cria três ícones vetoriais **DrawingImage**, adiciona os botões, um alternador Wrap e um botão suspenso Zoom com um **ContextMenu** do WPF, e um painel de texto e um indicador.
Images aceita qualquer **ImageSource**, tanto um **BitmapImage** quanto um **DrawingImage**. Quando a lista é uma **ObservableCollection<ImageSource>**, a barra de ferramentas é redesenhada quando uma imagem é adicionada, removida ou substituída.

Veja também

- Visão geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início rápido WinForms
- Início rápido Avalonia
- Imagens e glifos

Início rápido Avalonia

TsgcAvaloniaToolbar e **TsgcAvaloniaStatusBar** são as barras do Avalonia, os equivalentes de TsgcFMXToolbar e TsgcFMXStatusBar do Delphi, no Windows, no Linux e no macOS. Mouse, toque e caneta passam pelo mesmo caminho de ponteiro. O pacote é **esegece.sgcToolbar.Avalonia**.

Passos

- Crie um projeto .NET 8 e adicione os pacotes esegece.sgcToolbar.Avalonia, Avalonia.Desktop e Avalonia.Themes.Fluent.
- Mapeie o namespace com `xmlns:sgc="using:esegece.sgcToolbar"` e declare um **TsgcAvaloniaToolbar** e um **TsgcAvaloniaStatusBar**.
- No code-behind defina o **Theme**, adicione os itens e os painéis e atribua **OnItemClick**: é um delegate, não um evento do Avalonia.

Exemplo

MainWindow.axaml declara as duas barras e um texto.
MainWindow.axaml.cs define o tema escuro e adiciona um grupo de rádio Grid e List, um espaçador que empurra os itens seguintes para a direita, um rótulo, um botão Refresh, um painel de texto elástico e um painel de progresso.
Program.cs inicia o aplicativo com o tema Fluent.
Theme oculta a propriedade `StyledElement.Theme` dos controles do Avalonia, portanto defina-a em código em uma variável do tipo da barra, não como atributo XAML. Images é uma lista de **IImage**, um **Bitmap** ou um **DrawingImage**.

Veja também

- Visão geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início rápido WinForms
- Início rápido WPF
- Alternadores e grupos de rádio

Classes do .NET

Cada host tem uma classe de barra de ferramentas e uma classe de barra de status com os mesmos membros. Os tipos independentes de framework são compartilhados pelos três hosts e mantêm os nomes do Delphi.

Classes de host

Host	Classe	Classe 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 independentes de framework

Tipo	Descrição
TsgcToolbarModel	Os itens, o layout com a lista de excedentes, os itens em destaque, pressionados e com foco e o teste de acerto. Toda barra de ferramentas possui um, veja TsgcToolbarModel.
TsgcToolbarItems	A coleção de itens: Count, o indexador, Add, Insert, Delete, Clear, IndexOf, BeginUpdate e EndUpdate.
TsgcToolbarItem	Um item: Kind, Caption, Glyph, ImageIndex, Hint, Enabled, Visible, Down, GroupIndex, Width, Tag, DropdownMenu, OnClick, Click(), os resultados do layout BoundsRect, Shown e InOverflow, e na edição Registered Command e CommandParameter. Veja TsgcToolbarItem.
TsgcToolbarItemKind	tikButton, tikToggle, tikDropdown, tikSeparator, tikSpacer, tikLabel.
TsgcStatusBarModel	Os painéis, seu layout, a alça de redimensionamento e os setters thread-safe. Toda barra de status possui um, veja TsgcStatusBarModel.
TsgcStatusBarPanels	A coleção de painéis: Count, o indexador, Add, Insert, Delete, Clear, IndexOf, BeginUpdate e EndUpdate.
TsgcStatusBarPanel	Um painel: Kind, Text, Width, AutoSize, Spring, Alignment, Progress, IndicatorState, Hint, Tag e BoundsRect. Veja TsgcStatusBarPanel.
TsgcStatusPanelKind	spkText, spkProgress, spkIndicator.
TsgcStatusIndicatorState	sisOff, sisOk, sisWarning, sisError.

TAlignment	taLeftJustify, taRightJustify, taCenter: o alinhamento do texto de um painel, como o tipo da RTL do Delphi.
TsgcToolbarItemEvent	`void (object Sender, TsgcToolbarItem aItem)` , o tipo de OnItemClick.
TsgcStatusBarPanelEvent	`void (object Sender, TsgcStatusBarPanel aPanel)`, o tipo de OnPanelClick.
sgcToolbar_Render, sgcToolbar_Raster	O renderizador e a saída PNG headless, veja Renderização headless no .NET.

Barra de ferramentas

Membro	Tipo	Descrição
Model	TsgcToolbarModel	O modelo da barra de ferramentas.
Items	TsgcToolbarItems	Os itens, em ordem da esquerda para a direita.
ShowCaptions	bool	Desenha o texto dos botões ao lado de seu ícone. Sem isso, um botão com ícone mostra apenas o ícone.
ButtonSize, Spacing, Padding	int	A altura de um botão, o espaço entre os itens e o preenchimento ao redor deles, 32, 4 e 6 por padrão. A altura da barra de ferramentas segue ButtonSize e Padding.
Theme, AccentColor	TsgcUITheme, uint	O tema da barra. AccentColor 0 usa o destaque do tema.
Images	ImageList, IList<ImageSource>, IList<IImage>	As imagens de ImageIndex, veja Diferenças do .NET em relação ao Delphi.
OverflowMenu	ContextMenuStrip, ContextMenu	O menu do botão de excedentes, montado novamente a cada abertura. Somente na edição Registered.
TabStop (WinForms), Focusable (WPF, Avalonia)	bool	Permite que a barra receba o foco: Left, Right, Home e End movem o botão com foco, Enter e Space o ativam.
ItemAtPos(X, Y)	int	O índice do item em um ponto do controle, -1, ou CS_TOOLBAR_HIT_OVERFLOW sobre o botão de excedentes.
ItemRect(aIndex), OverflowRect()	Rectangle, Rect	O retângulo de um item ou do botão de excedentes nas coordenadas do controle.
ClickItem(aIndex)	void	Ativa um item como um clique faz.
OnItemClick	TsgcToolbarItemEvent	Um clique, Enter ou Space em um botão habilitado, depois que um alternador inverteu Down e antes de executar o OnClick ou o Command do item.

Barra de status

Membro	Tipo	Descrição

Model	TsgcStatusBarModel	O modelo da barra de status.
Panels	TsgcStatusBarPanels	Os painéis, em ordem da esquerda para a direita.
SizeGrip	bool	Desenha a alça de redimensionamento na extremidade direita e redimensiona a janela quando é arrastada, veja Diferenças do .NET em relação ao Delphi.
Theme, AccentColor	TsgcUITheme, uint	O tema da barra. AccentColor 0 usa o destaque do tema.
SetPanelText(aIndex, aText), SetProgress(aIndex, aValue), SetIndicator(aIndex, aState)	void	Definem o Text, o Progress (0 a 100) e o IndicatorState de um painel a partir de qualquer thread. Um índice fora do intervalo é ignorado.
PanelAtPos(X, Y)	int	O índice do painel em um ponto do controle, ou -1.
PanelRect(aIndex), GripRect()	Rectangle, Rect	O retângulo de um painel ou da alça de redimensionamento nas coordenadas do controle, vazio sem SizeGrip.
OnPanelClick	TsgcStatusBarPanelEvent	Um clique em um painel.
CS_STATUSBAR_HEIGHT	const int	A altura padrão, 24.

As posições e os tamanhos são pixels no WinForms (Rectangle) e unidades independentes de dispositivo no WPF e no Avalonia (Rect).

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia

Diferenças do .NET em relação ao Delphi

A API do .NET segue a API do Delphi nome por nome. Estes são os pontos em que ela difere.

Command em vez de action links

O Delphi vincula um item a uma TBasicAction por meio de sua propriedade Action, veja Action links. O .NET não tem TBasicAction, então o item tem **Command**, um `System.Windows.Input.ICommand`, e **CommandParameter**. Ambos fazem parte somente da edição Registered, como os action links do Delphi.

- `CanExecute(CommandParameter)` controla o **Enabled** do item: quando o comando é atribuído e a cada `CanExecuteChanged`.
- Um clique, Enter ou Space inverte um alternador, dispara o OnItemClick da barra e depois executa `Click()` do item: seu manipulador OnClick quando há um atribuído, caso contrário o comando, se ele puder ser executado.
- Caption, Hint e Down não são sincronizados a partir do comando, porque o ICommand não carrega nenhum deles. Continuam sendo propriedades do item, então um alternador mantém seu próprio estado Down. ImageIndex e Visible também não são sincronizados.
- Não há OnUpdate em tempo ocioso. Um comando que muda de estado dispara CanExecuteChanged, e o item o acompanha imediatamente.

Um programa de console que vincula um item Save a um comando mínimo. O item começa desabilitado, é habilitado quando o documento muda, e um clique salva e o desabilita novamente. Ele precisa apenas de esegece.sgcToolbar, as barras de ferramentas dos três hosts se comportam da mesma forma.

Menus suspensos e de excedentes

No Delphi **DropdownMenu** é um TPopupMenu da VCL ou do FireMonkey, veja Menus suspensos. No .NET ele é declarado como `object`, para que o modelo continue independente de framework, e cada host mostra abaixo do botão o tipo de menu de seu framework. **OverflowMenu** é montado pela barra de ferramentas a cada abertura, com uma entrada por item da lista de excedentes, e escolher uma entrada ativa o item como um clique faz, veja Excedentes.

Host	DropdownMenu	OverflowMenu
WinForms	System.Windows.Forms.ContextMenuStrip. Um menu descartado é removido de seu item.	ContextMenuStrip, o Hint de um item é o ToolTipText de sua entrada.
WPF	System.Windows.Controls.ContextMenu.	ContextMenu, com um Separator para um item separador e o Hint como o ToolTip de uma entrada.
Avalonia	Avalonia.Controls.ContextMenu.	ContextMenu, com um Separator para um item separador e o Hint como a dica de uma entrada.

O menu abre quando o botão é pressionado, depois do OnItemClick e do OnClick do item, então um manipulador pode preencher ou marcar o menu antes que ele apareça. O menu de excedentes não faz parte da edição Community: nela os itens que não cabem são cortados e a propriedade OverflowMenu não existe.

Imagens

O Delphi usa um TCustomImageList na VCL e um TCustomImageList de FMX.ImgList no FireMonkey, veja Imagens e glifos. No .NET a propriedade **Images** da barra de ferramentas tem o tipo de imagem de cada host, e o **ImageIndex** do item é um índice nela.

Host	Images	Desenhada em

Win-Forms	System.Windows.Forms.ImageList	O ImageSize da lista de imagens, em pixels.
WPF	IList<ImageSource>: bitmaps e ícones vetoriais DrawingImage. Os ícones vetoriais são rasterizados com o Skia, então também são desenhados em uma sessão de Área de Trabalho Remota minimizada ou desconectada. Uma ObservableCollection é redesenhada quando muda.	16 x 16 unidades independentes de dispositivo, nítida em qualquer DPI para ícones vetoriais.
Avalonia	IList<IImage>: um Bitmap ou um DrawingImage. Uma lista com INotifyCollectionChanged é redesenhada quando muda.	16 x 16 unidades independentes de dispositivo.

Cada imagem é convertida uma vez para ARGB e mantida em cache até a lista mudar, e no WPF e no Avalonia também até a escala de DPI mudar. Quando um item não tem imagem, o texto de seu **Glyph** é desenhado no espaço do ícone.

Dicas e tooltips

No Delphi as barras mostram o Hint do item ou do painel sob o cursor quando ShowHint é True, e o FMX mostra dicas a partir do 10.1 Berlin. No .NET as barras sempre mostram um tooltip, sem a propriedade ShowHint: o Hint do item sob o cursor, ou seu Caption quando o texto não é desenhado, e na barra de status o Hint do painel. O WinForms usa um componente ToolTip interno, o WPF define a propriedade ToolTip da barra e o Avalonia define a propriedade anexada ToolTip.Tip. Não defina você mesmo o tooltip de uma barra, a barra o substitui; defina o Hint do item ou do painel.

Atualizações de status thread-safe

Como no Delphi, **SetPanelText**, **SetProgress** e **SetIndicator** da barra de status podem ser chamados a partir de qualquer thread, veja Atualizações thread-safe. O modelo protege os valores com um lock e dispara seu OnChange na thread que chama, somente quando o valor mudou. A barra responde com uma solicitação de redesenho, e a partir de uma thread de trabalho ela nunca toca no controle:

- WinForms e WPF postam um callback por meio do `SynchronizationContext` que o controle capturou na thread da UI.
- O Avalonia posta um callback para `Dispatcher.UIThread`.
- Apenas um callback fica pendente por vez, então uma thread de trabalho que atualiza a barra milhares de vezes por segundo não inunda a fila da UI. O callback redesenha com os valores mais recentes.

Não é necessária nenhuma chamada a Invoke, BeginInvoke ou Dispatcher. Todas as outras propriedades, e adicionar, excluir ou configurar painéis, devem ser feitas a partir da thread da UI. O formulário abaixo baixa 50 partes simuladas em uma thread de trabalho e interrompe a thread antes de o formulário fechar.

Alça de redimensionamento

Com **SizeGrip** True, o padrão, a barra de status desenha os pontos da alça em sua extremidade direita e **GripRect** retorna seu retângulo, veja Alça de redimensionamento. No Delphi apenas a barra VCL redimensiona o formulário; a barra FMX apenas desenha a alça. No .NET a alça redimensiona a janela em todos os hosts, quando a janela é redimensionável e não está maximizada, e nunca em tempo de design:

Host	Como a alça redimensiona a janela
Win-Forms	A barra responde ao teste de acerto da área não cliente da alça com HTBOTTOMRIGHT, como a barra VCL, para um formulário com borda redimensionável.
WPF	Um pressionamento na alça inicia o laço de redimensionamento do sistema para a janela (WM_SYSCOMMAND com SC_SIZE), para uma Window com ResizeMode CanResize ou CanResizeWithGrip. O cursor muda sobre a alça.

Avalonia	Um pressionamento na alça chama BeginResizeDrag da janela, para uma Window com CanResize, no Windows, no Linux e no macOS.

Unidades

- WinForms: ButtonSize, Spacing, Padding e o Width de itens e painéis são pixels lógicos multiplicados por DPI / 96, como nas barras VCL com PPI / 96. ItemAtPos, ItemRect, PanelRect e GripRect são pixels do cliente.
- WPF e Avalonia: ItemAtPos, ItemRect, OverflowRect, PanelAtPos, PanelRect e GripRect são unidades independentes de dispositivo, como no FMX. O quadro e o layout dos modelos são pixels de dispositivo, as unidades multiplicadas pela escala de DPI do controle (WPF) ou pela escala de renderização da janela (Avalonia). O BoundsRect de um item ou de um painel também está em pixels de dispositivo.
- Headless: aWidth e aHeight são pixels lógicos e aScale os multiplica, veja Renderização headless no .NET.

Eventos

OnItemClick, OnPanelClick e o OnClick do item são propriedades delegate com os nomes do Delphi, não membros `event` do C#. Atribuir um manipulador substitui o anterior, como um ponteiro de método do Delphi, e eles são atribuídos em código, também no WPF e no Avalonia, onde o XAML não pode defini-los.

Sem verbos de designer

No Delphi as propriedades Items e Panels são editadas em tempo de design com o editor de coleções padrão. O port .NET não tem verbos de designer nem editor de coleções: os Items e Panels de uma barra não são serializados, então são sempre adicionados em código, normalmente no construtor do formulário ou da janela. No WinForms as barras estão na caixa de ferramentas e suas propriedades simples são definidas na janela Propriedades. No WPF e no Avalonia as barras são declaradas em XAML.

Edições

O port .NET é distribuído em pacotes Registered e Community, veja Visão geral do .NET. Os pacotes Community não têm OverflowMenu, Command e CommandParameter, como a edição Community do Delphi, veja Edições. Não há edição Trial.

Vinculação de dados

No WPF e no Avalonia, as propriedades visíveis ao usuário são propriedades de dependência e propriedades do Avalonia, portanto podem ser vinculadas, estilizadas e animadas, veja Vinculação de dados no .NET.

Veja também

- Visão geral do .NET
- Classes do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Action links
- Menus suspensos
- Imagens e glifos
- Atualizações thread-safe
- Alça de redimensionamento

Vinculação de dados no .NET

Nos hosts WPF e Avalonia, as propriedades visíveis ao usuário da barra de ferramentas e da barra de status são propriedades de dependência (WPF) e propriedades estilizadas do Avalonia, portanto podem ser vinculadas a um view model, definidas por um Style e animadas. Este tópico se aplica somente a WPF e Avalonia. Os controles WinForms não mudam: suas propriedades são propriedades CLR simples, como antes.

Propriedades vinculáveis

Cada propriedade da tabela tem um campo estático com o sufixo Property, por exemplo `TsgcWpfToolbar.ShowCaptionsProperty`. No WPF os campos são DependencyProperty e no Avalonia são StyledProperty. Nenhuma dessas propriedades é alterada pelo usuário, portanto todas são vinculadas em OneWay por padrão. Use `Mode=TwoWay` quando o view model também precisar receber um valor definido no controle pelo código.

Classe WPF	Classe Avalonia	Propriedades vinculáveis
TsgcWpfToolbar	TsgcAvaloniaToolbar	ShowCaptions, ButtonSize, Spacing, Padding, Theme, AccentColor
TsgcWpfStatusBar	TsgcAvaloniaStatusBar	Theme, SizeGrip, AccentColor

Padding é o preenchimento da barra em unidades independentes de dispositivo, um int. No WPF ele oculta o Padding do tipo Thickness de Control. No Avalonia, Theme é o tema do sgcUICore (TsgcUITheme) e oculta o tema de controle do Avalonia de StyledElement, que continua acessível por meio de `StyledElement.ThemeProperty`.

Comportamento

- **O modelo mantém os valores.** Cada propriedade é armazenada no modelo da barra de ferramentas ou da barra de status. Um valor vinculado vai para o modelo, e a propriedade retoma o valor que o modelo mantém: o modelo eleva ButtonSize para 8 e Spacing e Padding para 0, e a propriedade então mostra o valor em uso.
- **Um valor corrigido chega à origem.** Uma origem TwoWay também recebe o valor corrigido, no WPF e no Avalonia. O WPF não atualiza a origem enquanto transfere um valor dela, portanto no WPF o controle grava o valor corrigido de volta depois que o dispatcher executa o trabalho pendente. Uma alteração posterior feita no controle, pelo código ou no modelo, também chega a uma origem TwoWay nas duas plataformas. No Avalonia a gravação de volta cobre um valor que a origem envia quando a vinculação já está em execução. Um valor que a origem já tem quando a vinculação inicia é corrigido no controle, mas não é gravado de volta na origem.
- **As alterações feitas no modelo são copiadas de volta.** Quando o modelo altera um valor por conta própria, o controle o copia para a propriedade, de modo que as vinculações e os styles veem o valor em uso.
- **Styles.** Um setter de Style pode definir qualquer propriedade vinculável, por exemplo o Theme de todas as barras de ferramentas e barras de status de uma janela.

Propriedades que continuam sendo propriedades CLR

Propriedade	Motivo
Items (barra de ferramentas)	A coleção de botões do modelo, construída no código, e não um valor único para vincular.
Images (barra de ferramentas)	A lista de imagens dos botões, um objeto compartilhado por referência.

OverflowMenu (barra de ferramentas)	O menu que a barra de ferramentas cria para os botões que não cabem, somente leitura.
Panels (barra de status)	A coleção de painéis do modelo. Os valores dos painéis (Text, Progress, IndicatorState) são definidos pelos métodos thread-safe SetPanelText, SetProgress e SetIndicator, que podem ser chamados de qualquer thread.
Model	O modelo independente de plataforma por trás de cada controle, somente leitura.

Exemplo

O view model abaixo é compartilhado pelos dois exemplos. ShowCaptions e Theme são vinculados em OneWay, e ButtonSize é vinculado com `Mode=TwoWay`, de modo que um tamanho definido na barra de ferramentas pelo código volta ao view model. Os botões são adicionados no código, como nos inícios rápidos.

WPF

MainWindow.xaml cria o view model como DataContext da janela. O code-behind e o App.xaml são os do modelo WPF.

Avalonia

MainWindow.axaml define `x:DataType` como o view model, de modo que as vinculações são vinculações compiladas. O code-behind chama InitializeComponent, e a classe App é a de Início Rápido Avalonia.

Veja Também

- Diferenças do .NET em relação ao Delphi
- Classes do .NET
- Início Rápido WPF
- Início Rápido Avalonia

Renderização headless no .NET

Uma barra de ferramentas ou uma barra de status pode ser desenhada sem janela, em um servidor, em um teste ou em um relatório. **SaveToSVG** dos dois modelos grava um arquivo ou stream SVG apenas com o pacote esegece.sgcToolbar, e os métodos de extensão **SaveToPNG** de esegece.sgcToolbar.Rendering gravam um PNG desenhado no SkiaSharp, no Windows, no Linux e no macOS. A saída SVG é idêntica byte a byte à da renderização headless do Delphi.

Métodos

Método	Pacote	Descrição
TsgcToolbarModel.SaveToSVG e TsgcStatusBarModel.SaveToSVG(aFileName ou aStream, aWidth, aHeight, aScale = 1)	esegece.sgcToolbar	Grava a barra como SVG.
SaveToPNG(aFileName ou aStream, aWidth, aHeight, aScale = 1), um método de extensão dos dois modelos em sgcToolbar_Raster	esegece.sgcToolbar.Rendering	Grava a barra como PNG com fundo transparente.
TsgcToolbarModel.Render e TsgcStatusBarModel.Render(aCanvas, aWidth, aHeight, aScale = 1)	esegece.sgcToolbar	Desenha a barra em qualquer canvas do sgcUICore, uma barra de aWidth x aHeight pixels de dispositivo.
sgcToolbar_Render.RenderToolbar(aModel, aCanvas, aWidth, aHeight, aScale, aImages), sgcToolbar_Render.RenderStatusBar(aModel, aCanvas, aWidth, aHeight, aScale)	esegece.sgcToolbar	O renderizador por trás de Render.

Para SaveToSVG e SaveToPNG, aWidth e aHeight são pixels lógicos. A imagem tem aWidth x aScale por aHeight x aScale pixels, então uma escala de 2 dá uma imagem nítida para uma tela de alto DPI ou para impressão. O modelo da barra de ferramentas não tem lista de imagens: seu evento **OnGetImage**, `bool (int aIndex, out TsgcUIARGBBuffer aImage)`, retorna a imagem de um ImageIndex como um **TsgcUIARGBBuffer**; sem ele, o Glyph do item é desenhado no espaço do ícone.

Exemplo

Um programa de console (`net8.0`) que referencia esegece.sgcToolbar.Rendering e grava uma barra de ferramentas com um botão desabilitado, um alternador pressionado, um botão suspenso e um rótulo empurrado para a direita por um espaçador, e uma barra de status escura com um painel de texto elástico, um painel de progresso em 65 e um indicador de aviso, cada uma em PNG e SVG.

Veja também

- Visão geral do .NET
- Classes do .NET
- Renderização headless
- TsgcToolbarModel
- TsgcStatusBarModel

Demos do .NET

As demos do .NET estão na pasta `demos` do produto e no zip de demos, com a solução `sgcToolbar_Demos.sln`. Cada demo tem um README.md.

Demo	Host	Descrição
EditorShell	WinForms	A estrutura de um pequeno editor de texto com um TsgcWinFormsToolbar e um TsgcWinFormsStatusBar. A barra de ferramentas contém os botões New, Open e Save, os alternadores Bold e Italic, um grupo de rádio Left, Center e Right (GroupIndex 1), um botão suspenso Zoom com um menu de quatro níveis de zoom, um espaçador, um alternador de tema escuro e um botão Help. A barra de status mostra a linha e a coluna do cursor (AutoSize), um painel elástico com a última mensagem, o zoom e um indicador que fica amarelo enquanto o documento tem alterações não salvas. Save só fica habilitado enquanto há alterações não salvas, e estreitar a janela move os itens que não cabem para o menu de excedentes. Os botões usam os manipuladores OnClick dos itens e OnItemClick, então a demo compila com os pacotes Community e Registered.
EditorShellWpf	WPF	O mesmo editor em um TsgcWpfToolbar e um TsgcWpfStatusBar.
EditorShellAvalonia	Avalonia	O mesmo editor em um TsgcAvaloniaToolbar e um TsgcAvaloniaStatusBar, no Windows, no Linux e no macOS.
LiveStatus	WinForms	Uma barra de status ao vivo alimentada por uma thread de trabalho. Start executa uma thread que simula um download de 50 partes, sem rede, e informa seu progresso diretamente à barra de status com os métodos thread-safe SetPanelText, SetProgress e SetIndicator. A barra de status tem um painel de mensagem elástico, um painel de progresso, um indicador do feed (verde enquanto o download está em andamento, amarelo enquanto uma parte está lenta, vermelho quando está parado) e um contador de partes AutoSize. Stop encerra a thread, que informa seu término à thread da UI por meio do SynchronizationContext.
LiveStatusWpf	WPF	A mesma barra de status ao vivo em um TsgcWpfStatusBar.
LiveStatusAvalonia	Avalonia	A mesma barra de status ao vivo em um TsgcAvaloniaStatusBar.

Compilar e executar

As demos precisam do SDK do .NET 8 ou posterior. Execute uma demo a partir de sua pasta:

```
dotnet run
```

Dentro da árvore do produto, as demos compilam contra os projetos da biblioteca. A partir do zip de demos, elas restauram os pacotes NuGet, por padrão a edição Community. Defina **SgcToolbarEdition** para usar os pacotes Registered:

```
dotnet build -p:SgcToolbarEdition=Registered
```

Autoverificação

Cada demo tem uma opção `--test` que abre a janela fora da tela e a controla: EditorShell clica no grupo de rádio, nos alternadores, em Save, no menu Zoom, no menu de excedentes de uma janela estreita e no alternador Dark, e LiveStatus executa um download até o fim, interrompe um segundo na metade e troca o tema. A demo grava

uma linha PASS ou FAIL por verificação no arquivo e termina com o número de verificações que falharam, então 0 significa que todas passaram.

```
dotnet run -- --test result.txt
```

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Demos

Solução de problemas

Identificador não declarado Action ou OverflowMenu

A edição Community é compilada sem SGC_TOOLBAR_ADVANCED, portanto a propriedade Action dos itens e a propriedade OverflowMenu das barras de ferramentas não existem. O código que as usa falha com E2003 Undeclared identifier, e um formulário salvo por uma compilação Registered com um item vinculado a uma ação gera uma exceção "Error reading" ao ser carregado. Use o OnClick do item ou o evento OnItemClick da barra, ou uma compilação Registered ou Trial. Consulte Edições.

Os últimos itens são cortados na borda direita

Na edição Community não há menu de excedentes, portanto os itens que não cabem são cortados. Aumente a largura da barra, defina ShowCaptions como False ou use uma compilação Registered ou Trial. Consulte Excedentes.

O código após um menu é executado com atraso no FireMonkey

O TPopupMenu do FireMonkey de um item suspenso e do botão de excedentes é modal: Popup não retorna até que o menu seja fechado. O código colocado após ClickItem em um item suspenso, ou em OnItemClick depois que o menu foi exibido, só é executado quando o usuário fecha o menu. Faça o trabalho no OnClick das entradas do menu e não chame ClickItem em um item suspenso a partir de um timer ou de um teste que não pode esperar. No VCL, o menu abre ao pressionar o botão e seu laço modal captura a liberação do mouse.

A barra de ferramentas ignora o Height que defini

A altura de uma barra de ferramentas sempre segue ButtonSize mais duas vezes Padding. Altere essas duas propriedades.

Um alternador de rádio não pode ser desativado

Um alternador com GroupIndex maior que 0 é um botão de rádio: um clique só o ativa. Defina Down como False por código ou use GroupIndex 0 para um alternador independente. Consulte Alternadores e grupos de rádio.

Um item não reage a um clique

Um item desabilitado (Enabled False) é desenhado esmaecido e ignora cliques e o teclado, e separadores, espaçadores e rótulos nunca são clicados. Um item com uma Action segue o Enabled da ação, portanto verifique também o OnUpdate da ação.

A barra de status pisca ou gera erro quando atualizada a partir de uma thread

Somente SetPanelText, SetProgress e SetIndicator são thread-safe. Definir outras propriedades ou adicionar painéis a partir de uma thread de trabalho acessa o controle fora da thread principal. Consulte Atualizações thread-safe.

A alça de redimensionamento não redimensiona o formulário

No FMX a alça é apenas desenhada. No VCL ela só funciona quando o formulário tem uma borda redimensionável e não está maximizado. Consulte Alça de redimensionamento.

As imagens ou as dicas não aparecem no FireMonkey

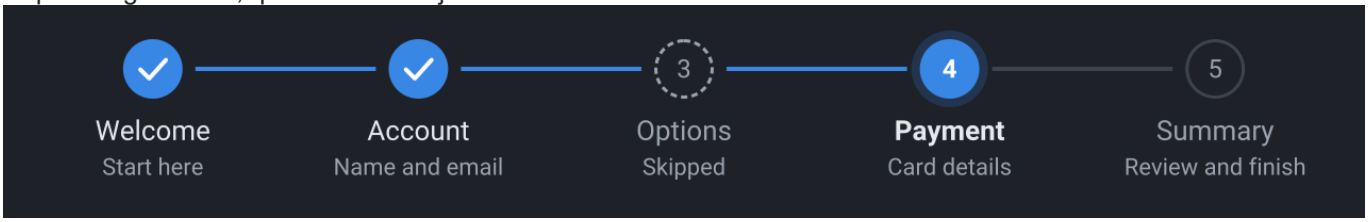
A propriedade Images da barra de ferramentas FMX requer o Delphi XE8 ou posterior, e as dicas das barras FMX requerem o 10.1 Berlin ou posterior com ShowHint True. Em versões mais antigas, os itens desenham o seu Glyph. Consulte Imagens e glifos.

O demo EditorShell ou EditorShellFMX não compila

Os dois demos usam Action links e o menu de excedentes, que não fazem parte da edição Community. Consulte Demos.

Introdução

sgcWizard adiciona assistentes passo a passo e indicadores de etapas a aplicações Delphi e C++Builder, para VCL e FireMonkey. **TsgcVCLWizard** e **TsgcFMXWizard** são contêineres de páginas: cada página é uma etapa, uma faixa de stepper mostra onde o usuário está e uma barra de botões navega entre as páginas com Back, Next, Finish e Cancel. **TsgcVCLStepper** e **TsgcFMXStepper** são os indicadores de etapas independentes, horizontais ou verticais, para qualquer fluxo que precise mostrar seu progresso.
O sgcWizard faz parte da suíte visual da eSeGeCe, ao lado do sgcDashboard, sgcNavigation, sgcGrid e sgcToolbar. Ele é desenhado com o **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro, de modo que um assistente tem a mesma aparência em VCL, em FireMonkey e na saída headless em PNG e SVG, e combina com os demais produtos da suíte. O sgcWizard requer o sgcUICore, que é instalado junto com ele.



Edições

O sgcWizard é distribuído nas edições Registered, Trial e Community. Todas as edições têm todos os recursos, e apenas a compilação Trial exibe um aviso de avaliação. Consulte Edições.

Por onde começar

- Arquitetura: o modelo do stepper independente de framework, o renderizador sobre o canvas do sgcUICore e os controles VCL e FMX.
- Plataformas: as versões do Delphi, o C++Builder e as plataformas.
- Início Rápido VCL, Início Rápido FMX e Início Rápido C++Builder: um primeiro assistente em poucos minutos.
- Componentes: os assistentes, as páginas, os steppers, as etapas e o modelo.
- Navegação e páginas ignoradas, Validação, Finish e Cancel, Teclado, Estados das etapas, Layout do stepper, Legendas dos botões, Temas, Tempo de design e Renderização headless.
- Demos: os projetos de demonstração para Delphi e C++Builder.

Arquitetura

O sgcWizard é construído em camadas. Cada camada depende apenas das que estão abaixo dela, de modo que o modelo e o renderizador compilam sem VCL nem FireMonkey.

Camada	Unit	Função
Modelo	sgcWizard_Model	TsgcStepperModel contém as Steps (uma coleção de TsgcWizardStep), a orientação, a etapa ativa, os estados automáticos, os tamanhos, o tema e a etapa em destaque. Seu método Layout calcula o círculo e o rótulo de cada etapa visível e os conectores entre elas, e HitTest encontra a etapa sob um ponto.
Renderizador	sgcWizard_Render	sgcWizardRender desenha um modelo em qualquer canvas do sgcUICore e sgcWizardRenderButtonBar desenha a barra de botões. sgcWizardSaveToPNG e sgcWizardSaveToSVG os utilizam para gravar uma imagem sem janela. Consulte Renderização headless.
Controles VCL	sgcWizard_VCL	TsgcVCLWizard, TsgcVCLWizardPage e TsgcVCLStepper. Somente Windows.
Controles FMX	sgcWizard_FMX	TsgcFMXWizard, TsgcFMXWizardPage e TsgcFMXStepper, com a mesma API e o mesmo comportamento.
sgcUICore	sgcUICore_Canvas, sgcUICore_VCL, sgcUICore_FMX, sgcUICore_Theme	O runtime compartilhado: o canvas vetorial e o rasterizador, a base de controles VCL e FMX, e os tokens dos temas claro e escuro com os aplicadores de tema.

Stepper

Um controle stepper possui um TsgcStepperModel e publica suas propriedades. A propriedade **Model** dá acesso ao restante: Layout, HitTest, IsStepClickable, IsConnectorDone, NextStep e PrevStep. Uma alteração no modelo agenda um novo quadro e nunca desenha diretamente, portanto muitas alterações seguidas custam uma única repintura.

Assistente e páginas

Um assistente é um contêiner de páginas, como o TPageControl com suas páginas TTabSheet. As páginas são controles filhos do assistente, na ordem dos controles, e somente a página ativa fica visível. Cada página contém controles comuns: edits, caixas de seleção, rótulos, frames ou qualquer outro controle.
O assistente possui dois controles filhos internos que não são serializados e não podem ser selecionados no designer: a **faixa do stepper**, um stepper cujas etapas espelham as páginas (Caption, Description, StepGlyph e Skip como estado ignorado), e a **barra de botões** com Back, Next (Finish na última página) e Cancel. O próprio assistente define os estados da faixa, portanto AutoStates da faixa é False. Ambos estão disponíveis em tempo de execução por meio das propriedades **Stepper** e **ButtonBar**.

Desenho

As etapas, os conectores e os botões são desenhados pelo renderizador. Todos os tamanhos são pixels lógicos multiplicados pela escala do monitor, de modo que os controles ficam nítidos em qualquer DPI.

Veja Também

- Plataformas
- TsgcStepperModel
- Temas

Edições

O sgcWizard é compilado a partir do mesmo código-fonte para todas as edições, e todas as edições têm todos os recursos.

Edição	Inclui
Registered	Tudo: os assistentes, páginas e steppers VCL e FMX, a validação, as páginas ignoradas, os estados das etapas, os temas, o suporte em tempo de design e a saída headless em PNG e SVG.
Trial	Os mesmos recursos da Registered. Uma compilação feita com SGC_TRIAL exibe uma mensagem informando que é uma versão de avaliação quando um componente é criado depois de passada a janela de lançamento indicada na compilação. Nada mais muda.
Community	Os mesmos recursos da Registered, sem aviso de avaliação.

A edição é uma escolha em tempo de compilação no sgcVer.inc. SGC_COMMUNITY também desativa SGC_TRIAL. Não é uma opção em tempo de execução, e o código escrito para uma edição compila nas demais.

Veja Também

- Plataformas
- Introdução

Instalação

O sgcWizard faz parte da distribuição sgcUI. Um único instalador cobre os produtos do sgcUI (sgcUICore, sgcCharts, sgcPDF, sgcReports, sgcMaps, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard) e instala apenas os produtos incluídos na sua licença.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcWizard** e quaisquer outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, na ordem de dependência, e instala os pacotes de design time.

O que acompanha o sgcWizard

sgcUICore, a base compartilhada dos produtos de layout e do sgcCharts, acompanha o sgcWizard. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcWizard é selecionado. O sgcWizard não tem pacote de ponte sgcWebSockets.

Pastas instaladas

Todos os produtos são instalados em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado. Este é o único library path do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e por versão da IDE.
`Lib<V>`	As units compiladas para cada versão da IDE e plataforma, por exemplo `LibD13`.
`Demos\Wizard`	Os demos do sgcWizard. Veja Demos.

O instalador adiciona `sgcUI\Source` ao library path e ao browsing path de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo trial, incluindo o sgcWizard.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcWizard. Os seus produtos são sgcCharts, sgcDashboard, sgcGrid, sgcNavigation, sgcToolbar e sgcWizard, enquanto o sgcPDF e o sgcMaps não fazem parte do Community. A compilação Community do sgcWizard tem os mesmos recursos que a edição Registered. Veja Edições.

Instalações autônomas anteriores

Se o instalador encontrar uma instalação autônoma mais antiga do sgcUICore, como a gravada pelo instalador autônomo do sgcCharts, ele lista os library paths e os pacotes de design dessa instalação na página **Previous installations**. Ao continuar, ele os remove da IDE antes de compilar os novos pacotes, de modo que nunca haja duas cópias de uma unit no caminho.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão da IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo de versão do Delphi, por exemplo `sgcWizardD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcWizard<V>` / `dclsgcWizard<V>`	Os componentes do sgcWizard (runtime / design time).

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcWizard`, depois `dclsgcWizard`. Em seguida, instale o pacote de design time `dclsgcWizard` e adicione `sgcUI\Source` ao library path da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcWizardC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

.NET

Nada muda para o .NET. O sgcWizard continua sendo distribuído como seus próprios pacotes NuGet (`esegece.sgcWizard` e os pacotes complementares). Veja a instalação do .NET.

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	TsgcVCLWizard, TsgcVCLWizardPage e TsgcVCLStepper no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	TsgcFMXWizard, TsgcFMXWizardPage e TsgcFMXStepper a partir do Delphi XE6 (SGC_WIZARD_FMX), no Windows, macOS, iOS e Android, e no Linux com FMX para Linux (FMXLinux). As dicas dos controles FMX exigem o 10.1 Berlin ou posterior.
Linux64	As units independentes de framework, o modelo e o renderizador com a saída headless em PNG e SVG, a partir do Delphi 10.2 (SGC_WIZARD_LINUX).
.NET	Um port em C# com a mesma API para WinForms, WPF e Avalonia, e a saída headless em PNG e SVG, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão Geral do .NET.

Pacotes

O sgcWizard traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcWizardD13.dpk para o Delphi 13 e sgcWizardC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcWizard	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, sgcUICore
sgcWizardVCL	Runtime, VCL	sgcWizard_VCL (TsgcVCLWizard, TsgcVCLWizardPage, TsgcVCLStepper)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcWizard
sgcWizardFMX	Runtime, FireMonkey, a partir do XE6	sgcWizard_FMX (TsgcFMXWizard, TsgcFMXWizardPage, TsgcFMXStepper)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcWizard
dclsgcWizard	Tempo de projeto, VCL	Registra TsgcVCLWizard e TsgcVCLStepper na página **SGC Wizard** da paleta, e o editor de componente que adiciona páginas.	designide, rtl, vcl, sgcWizard, sgcWizardVCL
dclsgcWizardFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXWizard e TsgcFMXStepper na página **SGC Wizard** da paleta, e o editor de componente que adiciona páginas.	designide, rtl, fmx, sgcWizard, sgcWizardFMX, dclsgcWizard

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcWizardFMX exige o dclsgcWizard, portanto instale primeiro o dclsgcWizard.

Versões do IDE e plataformas de destino

Delphi	sgcWizard	sgcWizardVCL	sgcWizardFMX	dclsgcWizard, dclsgcWizardFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcWizard, sgcWizardVCL	sgcWizardFMX	dclsgcWizard, dclsgcWizardFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcWizard)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcWizard)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcWizardC13.cbproj, sgcWizardVCLC13.cbproj, sgcWizardFMXC13.cbproj, dclsgcWizardC13.cbproj e dclsgcWizardFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXWizard, TsgcFMXWizardPage e TsgcFMXStepper) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcWizardVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcWizard_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcWizard é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa o modelo e o renderizador com a saída PNG e SVG sem interface. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.
As páginas não estão na paleta, elas são adicionadas com o verbo **Add Page**, consulte Tempo de design.

Veja Também

- Edições
- Arquitetura
- Solução de Problemas

Início Rápido VCL

Em tempo de design

- Coloque um **TsgcVCLWizard** da página **SGC Wizard** da paleta em um formulário e defina **Align** como alClient.
- Clique com o botão direito no assistente e escolha **Add Page** uma vez para cada etapa. Defina **Caption** e **Description** de cada página: elas são a legenda e a descrição da sua etapa.
- Coloque os controles de cada página nela. Escolha **Next Page** e **Previous Page** no menu de contexto, ou clique em uma etapa da faixa do stepper, para mostrar outra página no designer.
- Escreva um manipulador **OnBeforePageChange** que valide a página e defina aAllow como False quando a página não estiver completa, e um manipulador **OnFinish** que use os dados.
- Execute a aplicação: Next e Enter avançam, Back volta, Finish aparece na última página, e Cancel ou Escape disparam OnCancel.

Em código

O mesmo assistente criado em tempo de execução. O programa compila como está com `sgcUI\Source` no caminho de busca.

```
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;
```

Veja Também

- TsgcVCLWizard
- TsgcVCLWizardPage
- Validação
- Navegação e páginas ignoradas

Início Rápido FMX

Os controles FireMonkey exigem o Delphi XE6 ou posterior. Eles são usados exatamente como os controles VCL: coloque um **TsgcFMXWizard**, adicione as páginas com o verbo **Add Page**, coloque nelas os controles de cada página e trate **OnBeforePageChange** e **OnFinish**. As páginas são alinhadas como Client dentro do assistente. Um toque em um botão ou em uma etapa concluída funciona como um clique.

Em 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;
```

Veja Também

- TsgcFMXWizard

- TsgcFMXWizardPage
- Plataformas

Início Rápido C++Builder

Instale os pacotes do sgcWizard para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página **SGC Wizard** da paleta exatamente como no Delphi. O instalador adiciona os dois caminhos à IDE para você. Os cabeçalhos .hpp e os arquivos .lib gerados do sgcWizard e do sgcUICore estão nessa única pasta, portanto o sgcUICore não precisa de um caminho próprio.

Em código

OnBeforePageChange e OnFinish são eventos de closure, portanto seus manipuladores são funções membro __fastcall de uma classe. O parâmetro var aAllow é uma referência bool.

Veja Também

- TsgcVCLWizard
- Início Rápido VCL
- Demos

Demos

As demos estão na pasta **Demos** da instalação, com os projetos Delphi em Demos\Delphi e as versões C++Builder das demos VCL em Demos\CBuilder. Cada pasta de demo tem um readme.html que explica o que a demo faz e como ela funciona.

Demo	Componentes	O que ela mostra
Onboarding	TsgcVCLWizard, TsgcVCLWizardPage	Um assistente de conta com quatro páginas: Welcome, Account, Options e Summary. OnBeforePageChange verifica o nome e o e-mail da página Account e veta Next enquanto não forem válidos, uma caixa de seleção ignora a página Options, OnPageChanged preenche o resumo e OnFinish o exibe. Um seletor de tema claro e escuro e um combo de StepperPosition (topo, esquerda ou nenhum). Onboarding.exe /test result.txt conduz o assistente e grava uma linha PASS ou FAIL por verificação. Delphi e C++Builder.
Stepper	TsgcVCLStepper	Dois steppers independentes, um horizontal e um vertical, com as mesmas cinco etapas. Definir ActiveIndex conclui as etapas anteriores, botões marcam a etapa ativa como wssError ou wssSkipped, e um clique em uma etapa concluída move os dois steppers de volta a ela por meio de OnStepClick. Um seletor de tema claro e escuro. Delphi e C++Builder.
OnboardingFMX	TsgcFMXWizard, TsgcFMXWizardPage	O fluxo Onboarding em FireMonkey: a mesma validação, a mesma página Options ignorada, um resumo em Finish, uma caixa de seleção de tema escuro e as mesmas opções /test e /snapshot. Delphi.

As demos C++Builder incluem os cabeçalhos sgcWizard_*.hpp e sgcUICore_*.hpp e vinculam a biblioteca estática do pacote de runtime do C++Builder. Elas são compiladas com o compilador C++ clássico, Win32 e Win64.

Veja Também

- Início Rápido VCL
- Validação
- TsgcVCLStepper

TsgcVCLWizard

Unit: sgcWizard_VCL
TsgcVCLWizard é o assistente VCL, somente para Windows. Suas páginas são os controles filhos TsgcVCLWizardPage, a faixa do stepper mostra uma etapa por página e a barra de botões navega entre elas. Ele herda de TCustomControl e publica Align, Anchors, Constraints, Enabled, Font, ParentFont, ShowHint, ParentShowHint, TabOrder, TabStop e Visible.

Propriedades

Propriedade	Tipo	Descrição
ActivePageIndex	Integer	O índice da página visível. Defini-lo por código troca a página diretamente, sem eventos. Padrão 0.
ActivePage	classe de página	Pública. A página visível, ou nil. Defini-la troca a página sem eventos.
PageCount	Integer	Pública, somente leitura. O número de páginas.
Pages[Index]	classe de página	Pública, somente leitura. As páginas, na ordem dos controles.
StepperPosition	TsgcWizardStepperPosition	wspTop (padrão) mostra uma faixa de stepper horizontal no topo, wspLeft uma faixa vertical à esquerda, wspNone a oculta. Consulte Layout do stepper.
BackCaption, NextCaption, FinishCaption, CancelCaption	string	As legendas dos botões. Padrões Back, Next, Finish e Cancel. Elas só são gravadas quando alteradas. Consulte Legendas dos botões.
ShowCancel	Boolean	Mostra o botão Cancel. Padrão True.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. O assistente o repassa para a faixa, a barra de botões e os controles filhos das páginas. Consulte Temas.
AccentColor	TsgcUIColor	Um destaque $AARRGGBB para as etapas ativas e concluídas, os conectores concluídos e o botão padrão. 0 (padrão) usa o destaque do tema.
Stepper	classe de stepper	Pública, somente leitura. A faixa de stepper interna.
ButtonBar	classe de barra de botões	Pública, somente leitura. A barra de botões interna. ButtonCount, Buttons, ButtonRect e ButtonAt descrevem os botões como são desenhados.

Métodos

Método	Retorno	Descrição

Next	Boolean	Vai para a próxima página que não esteja ignorada, disparando OnBeforePageChange e OnPageChanged. Na última página chama Finish. True quando a página mudou ou o assistente foi concluído.
Back	Boolean	Volta para a página anterior que não esteja ignorada, com os mesmos eventos.
GoToPage(aIndex)	Boolean	Vai para qualquer página que não esteja ignorada, com os mesmos eventos. False para uma página ignorada ou um índice fora do intervalo.
Finish	Boolean	Dispara OnBeforePageChange com aToIndex = PageCount e, quando permitido, OnFinish.
Cancel		Dispara OnCancel.
NextPageIndex(aIndex), PrevPageIndex(aIndex)	Integer	A próxima ou anterior página após aIndex que não esteja ignorada, -1 quando nenhuma.
IsLastPage	Boolean	True quando nenhuma página após a ativa pode ser alcançada, de modo que Next mostra Finish.
IndexOfPage(aPage)	Integer	O índice de uma página, -1 quando ela não é uma página deste assistente.
AddPage(aCaption, aOwner = nil)	classe de página	Cria uma página com esse Caption, pertencente a aOwner (nil usa o proprietário do assistente), e a coloca por último.

Eventos

Evento	Tipo	Descrição
OnBeforePageChange	TsgcWizardPageChangeEvent	Antes de toda mudança por Next, Back, GoToPage, Finish ou um clique em uma etapa. aFromIndex, aToIndex e var aAllow: defina aAllow como False para permanecer na página. aToIndex = PageCount significa Finish. Consulte Validação.
OnPageChanged	TNotifyEvent	Depois que a página mudou.
OnFinish	TNotifyEvent	Depois que Finish foi permitido. Consulte Finish e Cancel.
OnCancel	TNotifyEvent	Cancel, o botão Cancel ou Escape.

Exemplo

```
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;
```

Veja Também

- TsgcVCLWizardPage
- TsgcFMXWizard
- Navegação e páginas ignoradas
- Teclado

TsgcVCLWizardPage

Unit: sgcWizard_VCL
TsgcVCLWizardPage é uma página de um TsgcVCLWizard. Ela herda de TCustomControl, aceita controles filhos e também publica Enabled, Font, ParentFont, Hint, ShowHint, ParentShowHint, PopupMenu, OnEnter, OnExit e OnResize. Uma página só pode ser colocada em um assistente: definir o seu Parent como qualquer outro controle gera EsgcWizardError.

Propriedades

Proprieda-de	Tipo	Descrição
Caption	string	A legenda da etapa da página.
Description	string	A descrição da etapa, desenhada abaixo da legenda enquanto ShowDescriptions da faixa for True, e a dica da etapa.
Skip	Boolean	Next e Back saltam a página e a faixa desenha a sua etapa como wssSkipped. Padrão False. Consulte Navegação e páginas ignoradas.
StepGlyph	string	Um texto curto desenhado no círculo da etapa. Vazio (padrão) desenha o número da etapa.
Visible	Boolean	Gerenciada pelo assistente: somente a página ativa fica visível. Não é gravada.
Wizard	classe de assis-tente	Pública, somente leitura. O assistente da página, nil quando ela não tem um.
PageIndex	Integer	Pública, somente leitura. A posição entre as páginas do assistente, -1 quando ela não está em um assistente.

O assistente desenha a página com a cor Surface do seu tema e aplica o tema aos controles filhos diretos da página, consulte Temas.

Exemplo

```
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;
```

Veja Também

- TsgcVCLWizard
- Tempo de design
- Solução de Problemas

TsgcFMXWizard

Unit: sgcWizard_FMX
TsgcFMXWizard é o assistente FireMonkey, a partir do Delphi XE6. Ele tem as mesmas propriedades, métodos e eventos do TsgcVCLWizard. As páginas são os filhos TsgcFMXWizardPage, alinhados como Client, a faixa do stepper é alinhada como MostTop ou MostLeft e a barra de botões como MostBottom. Posições e tamanhos são unidades lógicas do FMX. Ele publica Align, Anchors, CanFocus, ClipChildren, Enabled, Height, HitTest, Margins, Opacity, Padding, PopupMenu, Position, Size, TabOrder, TabStop, Visible e Width, ShowHint e ParentShowHint a partir do 10.1 Berlin, e OnEnter, OnExit, OnKeyDown, OnKeyUp e OnResize.

Propriedades

Propriedade	Tipo	Descrição
ActivePageIndex	Integer	O índice da página visível. Defini-lo por código troca a página diretamente, sem eventos. Padrão 0.
ActivePage	classe de página	Pública. A página visível, ou nil. Defini-la troca a página sem eventos.
PageCount	Integer	Pública, somente leitura. O número de páginas.
Pages[Index]	classe de página	Pública, somente leitura. As páginas, na ordem dos controles.
StepperPosition	TsgcWizardStepperPosition	wspTop (padrão) mostra uma faixa de stepper horizontal no topo, wspLeft uma faixa vertical à esquerda, wspNone a oculta. Consulte Layout do stepper.
BackCaption, NextCaption, FinishCaption, CancelCaption	string	As legendas dos botões. Padrões Back, Next, Finish e Cancel. Elas só são gravadas quando alteradas. Consulte Legendas dos botões.
ShowCancel	Boolean	Mostra o botão Cancel. Padrão True.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. O assistente o repassa para a faixa, a barra de botões e os controles filhos das páginas. Consulte Temas.
AccentColor	TsgcUIColor	Um destaque $AARRGGBB para as etapas ativas e concluídas, os conectores concluídos e o botão padrão. 0 (padrão) usa o destaque do tema.
Stepper	classe de stepper	Pública, somente leitura. A faixa de stepper interna.
ButtonBar	classe de barra de botões	Pública, somente leitura. A barra de botões interna. ButtonCount, Buttons, ButtonRect e ButtonAt descrevem os botões como são desenhados.

Métodos

Método	Retorno	Descrição

Next	Boolean	Vai para a próxima página que não esteja ignorada, disparando OnBeforePageChange e OnPageChanged. Na última página chama Finish. True quando a página mudou ou o assistente foi concluído.
Back	Boolean	Volta para a página anterior que não esteja ignorada, com os mesmos eventos.
GoToPage(aIndex)	Boolean	Vai para qualquer página que não esteja ignorada, com os mesmos eventos. False para uma página ignorada ou um índice fora do intervalo.
Finish	Boolean	Dispara OnBeforePageChange com aToIndex = PageCount e, quando permitido, OnFinish.
Cancel		Dispara OnCancel.
NextPageIndex(aIndex), PrevPageIndex(aIndex)	Integer	A próxima ou anterior página após aIndex que não esteja ignorada, -1 quando nenhuma.
IsLastPage	Boolean	True quando nenhuma página após a ativa pode ser alcançada, de modo que Next mostra Finish.
IndexOfPage(aPage)	Integer	O índice de uma página, -1 quando ela não é uma página deste assistente.
AddPage(aCaption, aOwner = nil)	classe de página	Cria uma página com esse Caption, pertencente a aOwner (nil usa o proprietário do assistente), e a coloca por último.

Eventos

Evento	Tipo	Descrição
OnBeforePageChange	TsgcWizardPageChangeEvent	Antes de toda mudança por Next, Back, GoToPage, Finish ou um clique em uma etapa. aFromIndex, aToIndex e var aAllow: defina aAllow como False para permanecer na página. aToIndex = PageCount significa Finish. Consulte Validação.
OnPageChanged	TNotifyEvent	Depois que a página mudou.
OnFinish	TNotifyEvent	Depois que Finish foi permitido. Consulte Finish e Cancel.
OnCancel	TNotifyEvent	Cancel, o botão Cancel ou Escape.

Exemplo

```
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;
```

Veja Também

- TsgcFMXWizardPage
- Início Rápido FMX
- Temas

TsgcFMXWizardPage

Unit: sgcWizard_FMX
TsgcFMXWizardPage é uma página de um TsgcFMXWizard, alinhada como Client dentro dele. Ela também publica ClipChildren, Enabled, HitTest, Padding, PopupMenu, OnEnter, OnExit e OnResize, e ShowHint e ParentShowHint a partir do 10.1 Berlin. Uma página só pode ser colocada em um assistente: definir o seu Parent como qualquer outro objeto gera EsgcWizardError.

Propriedades

Propriedade	Tipo	Descrição
Caption	string	A legenda da etapa da página.
Description	string	A descrição da etapa, desenhada abaixo da legenda enquanto ShowDescriptions da faixa for True, e a dica da etapa.
Skip	Boolean	Next e Back saltam a página e a faixa desenha a sua etapa como wssSkipped. Padrão False. Consulte Navegação e páginas ignoradas.
StepGlyph	string	Um texto curto desenhado no círculo da etapa. Vazio (padrão) desenha o número da etapa.
Visible	Boolean	Gerenciada pelo assistente: somente a página ativa fica visível. Não é gravada.
Wizard	classe de assistente	Pública, somente leitura. O assistente da página, nil quando ela não tem um.
PageIndex	Integer	Pública, somente leitura. A posição entre as páginas do assistente, -1 quando ela não está em um assistente.

Exemplo

```
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;
```

Veja Também

- TsgcFMXWizard
- TsgcVCLWizardPage
- Temas

TsgcVCLStepper

Unit: sgcWizard_VCL
TsgcVCLStepper é o indicador de etapas VCL, utilizável de forma independente. Ele mostra suas etapas como círculos numerados unidos por conectores, na horizontal ou na vertical. A dica de uma etapa é a sua Description. Enquanto o controle tem o foco, Left e Right (Up e Down na vertical) movem para a etapa habilitada visível anterior ou seguinte pelo mesmo veto de OnStepClick, enquanto Clickable for True. Ele também publica Align, Anchors, Constraints, Enabled, ShowHint, ParentShowHint, TabOrder, TabStop e Visible.

Propriedades

Propriedade	Tipo	Descrição
Steps	TsgcWizardSteps	As etapas.
Orientation	TsgcWizardOrientation	woHorizontal (padrão) ou woVertical.
ActiveIndex	Integer	A etapa ativa. Padrão 0. Defini-la por código não dispara OnActiveChange.
AutoStates	Boolean	Padrão True: definir ActiveIndex conclui as etapas visíveis anteriores, a menos que sejam wssError ou wssSkipped, e define as posteriores como wssPending. Consulte Estados das etapas.
ShowDescriptions	Boolean	Desenha a Description de cada etapa abaixo da sua legenda. Padrão True.
CircleSize	Integer	O diâmetro dos círculos das etapas, em pixels lógicos. Padrão 28.
ConnectorThickness	Integer	A largura dos conectores, em pixels lógicos. Padrão 2.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	Um destaque $AARRGGBB, 0 (padrão) usa o destaque do tema.
Clickable	Boolean	Um clique, ou as teclas de seta, pode ativar uma etapa concluída, ou uma etapa com erro anterior à ativa. Padrão True.
Model	TsgcStepperModel	Pública, somente leitura. O modelo do controle.
Wizard	classe de assistente	Pública, somente leitura. O assistente que hospeda o stepper como sua faixa, nil para um stepper independente.

Métodos

Método	Retorno	Descrição
StepAt(X, Y)	Integer	A etapa sob um ponto do cliente, -1 quando nenhuma.

StepRect(aIndex)	TRect (TRectF no FMX)	O círculo e o rótulo de uma etapa em coordenadas do cliente, vazio quando a etapa não é exibida.

Eventos

Evento	Tipo	Descrição
OnStep-Click	TsgcWizardS-tepClickEvent	Uma etapa clicável foi clicada, ou uma tecla de seta se move para ela. aIn-dex e var aAllow: defina aAllow como False para manter a etapa ativa.
OnActive-Change	TNotifyEvent	A etapa ativa mudou por um clique ou uma tecla. Nunca é disparado quan-do ActiveIndex é definido por código.

Exemplo

```
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;
```

Veja Também

- TsgcFMXStepper
- Estados das etapas
- Layout do stepper

TsgcFMXStepper

Unit: sgcWizard_FMX
TsgcFMXStepper é o indicador de etapas FireMonkey, a partir do Delphi XE6, com as mesmas propriedades, métodos e eventos do TsgcVCLStepper. A partir do 10.1 Berlin, o Hint do controle segue a Description da etapa sob o mouse. Ele também publica Align, Anchors, CanFocus, Enabled, Height, HitTest, Margins, Opacity, Padding, PopupMenu, Position, Size, TabOrder, TabStop, Visible e Width, e OnEnter, OnExit, OnKeyDown, OnKeyUp, OnMouseDown, OnMouseMove, OnMouseUp e OnResize.

Propriedades

Propriedade	Tipo	Descrição
Steps	TsgcWizardSteps	As etapas.
Orientation	TsgcWizardOrientation	woHorizontal (padrão) ou woVertical.
ActiveIndex	Integer	A etapa ativa. Padrão 0. Defini-la por código não dispara OnActiveChange.
AutoStates	Boolean	Padrão True: definir ActiveIndex conclui as etapas visíveis anteriores, a menos que sejam wssError ou wssSkipped, e define as posteriores como wssPending. Consulte Estados das etapas.
ShowDescriptions	Boolean	Desenha a Description de cada etapa abaixo da sua legenda. Padrão True.
CircleSize	Integer	O diâmetro dos círculos das etapas, em pixels lógicos. Padrão 28.
ConnectorThickness	Integer	A largura dos conectores, em pixels lógicos. Padrão 2.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	Um destaque $AARRGGBB, 0 (padrão) usa o destaque do tema.
Clickable	Boolean	Um clique, ou as teclas de seta, pode ativar uma etapa concluída, ou uma etapa com erro anterior à ativa. Padrão True.
Model	TsgcStepperModel	Pública, somente leitura. O modelo do controle.
Wizard	classe de assistente	Pública, somente leitura. O assistente que hospeda o stepper como sua faixa, nil para um stepper independente.

Métodos

Método	Retorno	Descrição
StepAt(X, Y)	Integer	A etapa sob um ponto do cliente, -1 quando nenhuma.

StepRect(aIndex)	TRect (TRectF no FMX)	O círculo e o rótulo de uma etapa em coordenadas do cliente, vazio quando a etapa não é exibida.

Eventos

Evento	Tipo	Descrição
OnStep-Click	TsgcWizardS-tepClickEvent	Uma etapa clicável foi clicada, ou uma tecla de seta se move para ela. aIn-dex e var aAllow: defina aAllow como False para manter a etapa ativa.
OnActive-Change	TNotifyEvent	A etapa ativa mudou por um clique ou uma tecla. Nunca é disparado quan-do ActiveIndex é definido por código.

Exemplo

```
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;
```

Veja Também

- TsgcVCLStepper
- Estados das etapas
- Plataformas

TsgcWizardStep

Unit: sgcWizard_Model
TsgcWizardStep é uma etapa de um stepper, um item de TsgcWizardSteps. Em um assistente, as etapas da faixa são criadas a partir das páginas, portanto a aplicação define as propriedades da página.

Propriedades

Propriedade	Tipo	Descrição
Caption	string	A legenda desenhada abaixo do círculo (horizontal) ou à sua direita (vertical).
Description	string	A segunda linha do rótulo, desenhada enquanto ShowDescriptions for True, e a dica da etapa.
Glyph	string	Um texto curto desenhado no círculo. Vazio desenha o número da etapa.
State	TsgcWizardStepState	wssPending (padrão), wssActive, wssCompleted, wssError ou wssSkipped. Consulte Estados das etapas.
Enabled	Boolean	Uma etapa desabilitada é desenhada com cores esmaecidas e não pode ser clicada. Padrão True.
Visible	Boolean	Uma etapa oculta não recebe espaço nem número, e mantém o seu estado. Padrão True.
Tag	NativeInt (Integer antes do XE2)	Livre para a aplicação. Padrão 0.
Shown, Number, CircleRect, LabelRect	Boolean, Integer, TsgcUIRect	Públicas, somente leitura. Os resultados do último Layout: se a etapa recebeu um espaço, o seu número, começando em 1, entre as etapas visíveis, e os retângulos do seu círculo e do seu rótulo em pixels do dispositivo.

Métodos

Método	Retorno	Descrição
CircleText	string	O texto desenhado no círculo: Glyph, ou Number quando Glyph está vazio.

Exemplo

```
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;
```

Veja Também

- TsgcWizardSteps
- TsgcStepperModel

TsgcWizardSteps

Unit: sgcWizard_Model
TsgcWizardSteps é a coleção própria de itens TsgcWizardStep por trás da propriedade Steps de um stepper e de TsgcStepperModel. Ela é editada em tempo de design com o editor de coleções padrão.

Métodos

Método	Retorno	Descrição
Add	TsgcWizardStep	Adiciona uma etapa vazia.
AddStep(aCaption, aDescription = '', aGlyph = '')	TsgcWizardStep	Adiciona uma etapa com a sua legenda, descrição e glifo e a retorna.
Items[Index]	TsgcWizardStep	A propriedade indexada padrão.

Uma etapa adicionada ou removida recalcula os estados automáticos enquanto AutoStates for True.

Exemplo

```
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;
```

Veja Também

- TsgcWizardStep
- TsgcVCLStepper

TsgcStepperModel

Unit: sgcWizard_Model
TsgcStepperModel é o modelo de um stepper independente de framework: as etapas com seu estado, a orientação, a etapa ativa, a etapa em destaque e o layout dos círculos, rótulos e conectores com o seu teste de acerto. Os steppers VCL e FMX possuem um e publicam suas propriedades, e a renderização headless o desenha sem janela. Ele deve ser usado a partir da thread principal.

Propriedades

Propriedade	Tipo	Descrição
Steps, Orientation, ActiveIndex, AutoStates, ShowDescriptions, CircleSize, ConnectorThickness, Theme, AccentColor, Clickable		As propriedades publicadas dos steppers, consulte TsgcVCLStepper.
HotIndex	Integer	A etapa sob o mouse, definida pelo controle. Uma etapa em destaque clicável recebe uma sobreposição sutil.
ConnectorCount, Connectors[Index]	Integer, TsgcWizardConnector	Os conectores do último Layout: FromIndex, ToIndex e a linha X1, Y1, X2, Y2 em pixels do dispositivo.
LayoutWidth, LayoutHeight, LayoutScale	Integer, Double	O tamanho e a escala do último Layout.
OnChange	TNotifyEvent	Qualquer alteração do modelo. Layout nunca o dispara.

Métodos

Método	Retorno	Descrição
Layout(aWidth, aHeight, aScale = 1) e sobrecargas com um callback de medição ou um canvas		Calcula os retângulos das etapas e dos conectores para um stepper de aWidth por aHeight pixels do dispositivo. Todo tamanho lógico é multiplicado por aScale. Sem um callback de medição ou um canvas, as larguras dos rótulos são estimadas.
HitTest(aX, aY)	Integer	A etapa cujo círculo ou rótulo contém o ponto do último Layout, -1 quando nenhuma. As etapas desabilitadas também são informadas.
IsStepClickable(aIndex)	Boolean	Enquanto Clickable, uma etapa visível e habilitada que não seja a ativa é clicável quando é wssCompleted, ou wssError antes da etapa ativa.
IsConnectorDone(aToIndex)	Boolean	A regra do conector, consulte Estados das etapas.

ApplyAutoStates		Aplica os estados automáticos agora, seja qual for o valor de AutoStates.
VisibleCount	Integer	O número de etapas visíveis.
NextStep(aIndex), PrevStep(aIndex)	Integer	A próxima ou anterior etapa visível e habilitada, -1 quando nenhuma.
GetAccent	TsgcUIColor	O destaque em uso: AccentColor, ou o destaque do tema quando for 0.

Exemplo

O programa de console abaixo 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;
```

Veja Também

- TsgcWizardStep
- Renderização headless
- Estados das etapas

Navegação e Páginas Ignoradas

O usuário se move por um assistente com a barra de botões, o teclado e as etapas concluídas da faixa do stepper. A aplicação usa os mesmos métodos.

Ação	Efeito
Next, o botão Next, Enter	Avança para a próxima página que não esteja ignorada. Na última página chama Finish.
Back, o botão Back	Volta para a página anterior que não esteja ignorada. O botão Back fica desabilitado quando nenhuma página anterior pode ser alcançada.
GoToPage(aIndex)	Vai para qualquer página que não esteja ignorada.
Um clique em uma etapa concluída da faixa	Chama GoToPage para a sua página.
ActivePageIndex ou ActivePage definidos por código	Troca a página diretamente, sem eventos.

Todo movimento, exceto a troca direta, dispara OnBeforePageChange, que pode vetá-lo, e depois OnPageChanged. Quando o foco estava dentro do assistente, uma mudança de página o move para o primeiro controle da nova página, ou para o assistente.

Páginas ignoradas

Uma página cuja propriedade **Skip** é True permanece no assistente, mas Next e Back a saltam e a faixa do stepper desenha a sua etapa como ignorada, com um contorno tracejado. Skip pode mudar a qualquer momento, por exemplo a partir de uma caixa de seleção em uma página anterior. NextPageIndex e PrevPageIndex retornam a página para a qual Next e Back iriam, e IsLastPage é True quando nenhuma página após a ativa pode ser alcançada, de modo que o botão Next exibe a legenda Finish.

Exemplo

```
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;
```

Veja Também

- TsgcVCLWizard
- Validação
- Finish e Cancel

Validação

OnBeforePageChange dispara antes de toda mudança de página feita por Next, Back, GoToPage, Finish ou um clique em uma etapa. Ele recebe aFromIndex, a página ativa, aToIndex, a página que está prestes a ser exibida, e um var aAllow que começa como True. Defina aAllow como False para permanecer na página, por exemplo enquanto um campo obrigatório está vazio.

- **aToIndex > aFromIndex** é todo movimento para frente, Finish incluído, de modo que um único teste valida a página seja qual for o botão pressionado pelo usuário.
- **aToIndex = PageCount** significa que o usuário pressionou Finish: Finish dispara o evento com PageCount como destino, que nunca é um índice de página real.
- Os movimentos para trás geralmente não precisam de validação. Teste aToIndex < aFromIndex para deixá-los passar.
- Quando a mudança é vetada, nada mais acontece: a página, a faixa e os botões permanecem como estão, portanto mostre o motivo e mova o foco para o campo.

Exemplo

```
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;
```

Veja Também

- Navegação e páginas ignoradas
- Finish e Cancel
- Demos

Finish e Cancel

Na última página, o botão Next exibe o **FinishCaption** e é o botão padrão. Pressioná-lo, pressionar Enter ou chamar **Finish** dispara OnBeforePageChange com aToIndex = PageCount e, quando isso é permitido, **OnFinish**. O assistente não fecha nada por conta própria: OnFinish decide o que acontece, por exemplo salvar os dados e fechar o formulário, ou definir o ModalResult de uma caixa de diálogo.
O botão Cancel, a tecla Escape e uma chamada a **Cancel** disparam **OnCancel**. Ele não tem veto e não muda a página. Defina **ShowCancel** como False para ocultar o botão Cancel.

Exemplo

```
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;
```

Veja Também

- Validação
- Teclado
- Legendas dos botões

Teclado

Enquanto o assistente ou um controle dentro dele tem o foco, **Enter** é Next (Finish na última página) e **Escape** é Cancel. Um controle que quer a tecla para si tem prioridade: um TMemo mantém o Enter para uma nova linha, e um botão com foco, ou um TButton com Default ou Cancel definido, trata a tecla normalmente.

Framework	Como as teclas chegam
VCL	Por CM_DIALOGKEY a partir do formulário pai, ou por KeyDown quando o próprio assistente tem o foco.
FireMonkey	Pela cadeia de teclas de diálogo do formulário (DialogKey e AfterDialogKey), ou por KeyDown quando o próprio assistente tem o foco.

Após uma mudança de página, o foco vai para o primeiro controle da nova página, ou para o assistente, quando estava dentro do assistente antes.

Teclas do stepper

Um stepper com foco se move com Left e Right, ou Up e Down quando é vertical, para a etapa habilitada visível anterior ou seguinte. O movimento passa pelo mesmo veto de OnStepClick que um clique e só acontece enquanto Clickable for True.

Veja Também

- Navegação e páginas ignoradas
- Finish e Cancel
- TsgcVCLStepper

Estados das Etapas

Toda etapa tem um **State** que decide como ela é desenhada.

Estado	Desenhado como
wssPending	Um contorno na cor da borda com o número ou glifo da etapa na cor de texto suave.
wssActive	Um disco de destaque com um halo suave de destaque e o texto da etapa em uma cor de contraste. A legenda fica em negrito.
wssCompleted	Um disco de destaque com uma marca de verificação.
wssError	Um disco vermelho com um ponto de exclamação e uma legenda vermelha.
wssSkipped	Um contorno tracejado com o texto da etapa na cor de texto suave.

Uma etapa desabilitada é desenhada com cores esmaecidas.

Estados automáticos

Enquanto **AutoStates** é True (padrão), cada vez que ActiveIndex é definido, uma etapa é adicionada ou removida, ou AutoStates é ativado, as etapas visíveis anteriores à ativa passam a wssCompleted, a menos que sejam wssError ou wssSkipped, a etapa ativa passa a wssActive e as etapas visíveis posteriores passam a wssPending. As etapas ocultas mantêm o seu estado. Com AutoStates False, todos os estados ficam a cargo da aplicação. Em um assistente, a faixa tem AutoStates False, porque o assistente define os estados a partir das páginas e da sua propriedade Skip.

A regra do conector

O conector entre duas etapas visíveis consecutivas está concluído, desenhado na cor de destaque, quando a etapa para a qual ele leva é wssCompleted ou wssActive, ou quando essa etapa é wssSkipped ou wssError e já foi ultrapassada, isto é, alguma etapa visível posterior é wssActive ou wssCompleted. Caso contrário, ele é desenhado na cor da borda. **Model.IsConnectorDone(aToIndex)** responde à regra para o conector que leva à etapa aToIndex, de modo que uma página ignorada no meio de uma parte concluída do fluxo não interrompe a linha de destaque.

Exemplo

```
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;
```

Veja Também

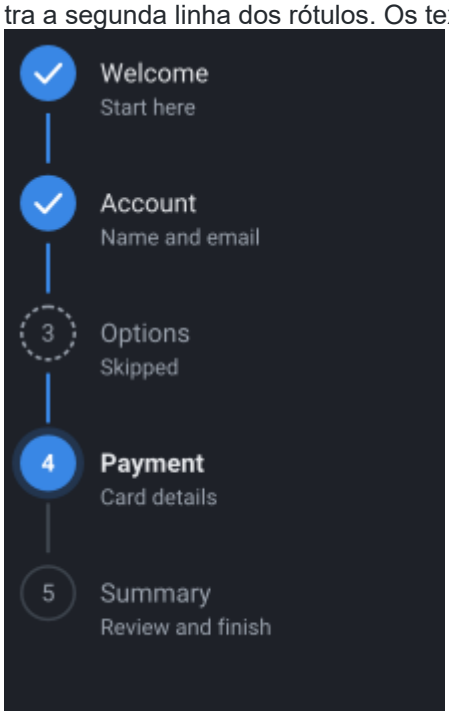
- TsgcWizardStep
- TsgcStepperModel
- Layout do stepper

Layout do Stepper

Um stepper é horizontal ou vertical, definido por **Orientation**. Somente as etapas visíveis recebem um espaço, e o seu número é a sua posição, começando em 1, entre as etapas visíveis.

- **woHorizontal**: a largura é dividida em espaços iguais, o círculo fica centralizado no seu espaço, e a legenda e a descrição ficam centralizadas abaixo dele. O bloco é centralizado verticalmente.
- **woVertical**: a altura é dividida em linhas iguais, o círculo fica no canto superior esquerdo da sua linha, e o rótulo fica à sua direita com a legenda centralizada no círculo. Os conectores descem entre os círculos.

CircleSize (padrão 28) e **ConnectorThickness** (padrão 2) são pixels lógicos. **ShowDescriptions** oculta ou mostra a segunda linha dos rótulos. Os textos dos rótulos que não cabem são cortados no espaço disponível.



StepperPosition de um assistente

Um assistente posiciona a sua faixa com **StepperPosition**: wspTop (padrão) mostra uma faixa horizontal no topo, com 80 pixels lógicos de altura, wspLeft uma faixa vertical à esquerda, com 200 pixels lógicos de largura, e wspNone oculta a faixa. A barra de botões na parte inferior tem 52 pixels lógicos de altura.

Exemplo

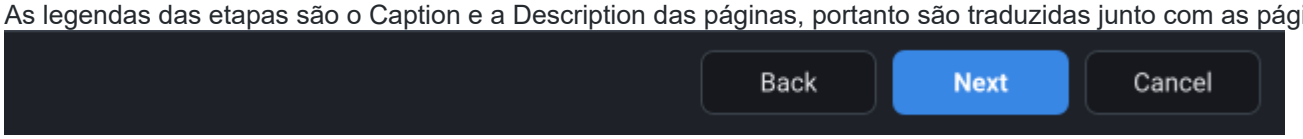
```
uses
  SysUtils, Forms, Classes, Controls, StdCtrls, Dialogs,
  sgcUICore_Theme, sgcWizard_Model, sgcWizard_VCL;

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;
```

Veja Também

- Estados das etapas
- TsgcVCLStepper
- TsgcVCLWizard

Legendas dos Botões e Localização

As quatro legendas da barra de botões são propriedades publicadas do assistente: **BackCaption**, **NextCaption**, **FinishCaption** e **CancelCaption**, com os valores padrão Back, Next, Finish e Cancel. Uma legenda só é gravada no arquivo do formulário quando difere do padrão, portanto um formulário com as legendas padrão assume novos padrões, e as legendas podem ser traduzidas com as ferramentas de recursos habituais da IDE ou definidas em código a partir da tradução da aplicação. Os botões crescem junto com suas legendas, e **ShowCancel** oculta o botão Cancel.
As legendas das etapas são o Caption e a Description das páginas, portanto são traduzidas junto com as páginas.



Exemplo

```
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;
```

Veja Também

- Finish e Cancel
- TsgcVCLWizard

Temas

Theme define as cores a partir dos tokens de tema do sgcUICore: utLight (padrão), utDark ou utCustom. **AccentColor** substitui o destaque do tema por uma cor $AARRGGBB, usada pelas etapas ativas e concluídas, pelos conectores concluídos e pelo botão padrão. 0 (padrão) mantém o destaque do tema.
Um assistente repassa Theme e AccentColor para a sua faixa de stepper e a sua barra de botões, desenha a si mesmo e às suas páginas com a cor Surface do tema, e aplica o tema aos controles filhos diretos de cada página: em uma mudança de tema, quando o assistente é carregado e quando um controle é adicionado a uma página. Um controle filho com sua própria propriedade Theme recebe o tema. Qualquer outro filho recebe a cor Surface como fundo e a cor Text como cor da fonte, de modo que um TLabel, um TEdit ou um TCheckBox continue legível em uma página escura.

- **FireMonkey**: os controles edit estilizados, como TEdit, TMemo, TComboBox, TListBox, os controles de data e as grades, mantêm as cores do seu estilo, porque o estilo desenha um fundo opaco próprio. A cor do texto deles só é alterada quando a cor de fundo também é definida.
- **utCustom** deixa os controles filhos intocados. Na VCL as páginas assumem a cor do seu pai, no FireMonkey nem o assistente nem as páginas são desenhados.

Os tokens de tema são os mesmos do sgcDashboard e dos demais produtos do sgcUICore, portanto um assistente e um dashboard com o mesmo Theme combinam, e um assistente hospedado em um tile do sgcDashboard segue o tema do dashboard.

Exemplo

```
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;
```

Veja Também

- Arquitetura
- Layout do stepper
- Renderização headless

Tempo de Design

O pacote de tempo de design registra TsgcVCLWizard, TsgcVCLStepper, TsgcFMXWizard e TsgcFMXStepper na página **SGC Wizard** da paleta. As páginas não estão na paleta. O editor de componentes do assistente e de suas páginas adiciona três verbos ao menu de contexto:

Verbo	Efeito
Add Page	Cria uma nova página com o nome derivado da sua classe, usando esse nome como Caption, coloca-a por último no assistente, a exibe e a seleciona.
Next Page	Mostra a próxima página no designer.
Previous Page	Mostra a página anterior no designer.

Um clique em uma etapa da faixa do stepper no designer também mostra a sua página. Somente a página ativa fica visível, e a página exibida no designer é o ActivePageIndex gravado no arquivo do formulário. A faixa do stepper e a barra de botões são internas: elas não são gravadas no arquivo do formulário e não podem ser selecionadas. As Steps de um stepper independente são editadas com o editor de coleções padrão.

Veja Também

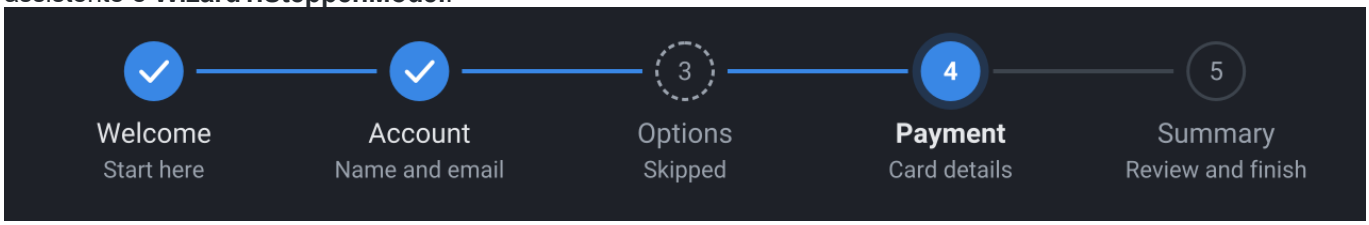
- Início Rápido VCL
- TsgcVCLWizardPage
- Plataformas

Renderização Headless

O renderizador em sgcWizard_Render desenha um TsgcStepperModel e a barra de botões sem janela e sem VCL nem FMX, de modo que uma aplicação de console, um serviço, um teste ou um servidor web possa produzir o mesmo stepper como imagem, por exemplo para a documentação de uma aplicação ou um teste visual. A unit compila em todas as plataformas, inclusive Linux64.

Rotina	Descrição
sgcWizardSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	Grava um arquivo ou stream PNG. aWidth e aHeight são pixels lógicos e aScale os multiplica: 640 por 100 na escala 2 resulta em uma imagem de 1280 por 200.
sgcWizardSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	Grava um arquivo ou stream SVG do mesmo stepper.
sgcWizardRender(aModel, aCanvas, aWidth, aHeight, aScale = 1)	Desenha o stepper em qualquer canvas do sgcUICore, em pixels do dispositivo.
sgcWizardSaveButtonBarToPNG, sgcWizardSaveButtonBarToSVG (aButtons, aStream, aWidth, aHeight, aScale = 1, aTheme = utLight, aAccent = 0)	Grava uma barra de botões. sgcWizardButton(aCaption, aEnabled, aIsDefault, aHot, aPressed) cria um botão, e um botão com legenda vazia fica oculto.
sgcWizardRenderButtonBar, sgcWizardButtonBarLayout, sgcWizardButtonBarHitTest	Desenham, dispõem e testam o acerto de uma barra de botões em qualquer canvas do sgcUICore.

O modelo de um controle stepper está disponível por meio da sua propriedade **Model**, de modo que **sgcWizardSaveToPNG(Stepper1.Model, ...)** salva o que o controle mostra em qualquer tamanho. A faixa de um assistente é **Wizard1.Stepper.Model**.



Exemplo

```
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;
```

Veja Também

- TsgcStepperModel
- Plataformas
- Estados das etapas

Visão Geral do .NET

O **sgcWizard .NET** é um port gerenciado em C# da biblioteca Delphi sgcWizard, não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcWizard** e mantêm os nomes do Delphi, então **TsgcStepperModel**, **TsgcWizardSteps** e **TsgcWizardStepState** se leem da mesma forma nas duas linguagens, e o assistente, a página e o stepper de cada host seguem o padrão de nomes do Delphi. Ele é desenhado com o port .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual eSeGeCe, de modo que um assistente tem a mesma aparência no WinForms, no WPF, no Avalonia e na saída headless em PNG e SVG.

Pacotes

O port é dividido em cinco pacotes NuGet, para que um projeto inclua apenas o host que utiliza.

PackageId	Contém	Frameworks de destino
esegece.sgcWizard	O modelo do stepper (**TsgcStepperModel**, **TsgcWizardSteps**, **TsgcWizardStep**), o renderizador **sgcWizard_Render** e a saída headless em SVG. Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcWizard.Rendering	A saída headless em PNG, **sgcWizard_Render_Raster**, desenhada no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, Linux e macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcWizard.WinForms	**TsgcWinFormsWizard**, **TsgcWinFormsWizardPage**, **TsgcWinFormsStepper** e **TsgcWinFormsWizardButtonBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcWizard.Wpf	**TsgcWpfWizard**, **TsgcWpfWizardPage**, **TsgcWpfStepper** e **TsgcWpfWizardButtonBar**.	net472, net8.0-windows, net9.0-windows
esegece.sgcWizard.Avalonia	**TsgcAvaloniaWizard**, **TsgcAvaloniaWizardPage**, **TsgcAvaloniaStepper** e **TsgcAvaloniaWizardButtonBar**, no Windows, Linux e macOS com o Avalonia 11.	net8.0, net9.0

Os três pacotes de host dependem de esegece.sgcWizard.Rendering, que depende de esegece.sgcWizard, então uma única linha `PackageReference` traz tudo o que um host precisa. Veja Instalação via NuGet.

Edições

Edição	Ids dos pacotes	Inclui
Registered	esegece.sgcWizard, esegece.sgcWizard.Rendering, esegece.sgcWizard.WinForms, esegece.sgcWizard.Wpf, esegece.sgcWizard.Avalonia	Tudo: os assistentes, as páginas, os steppers e as barras de botões dos três hosts, a validação, as páginas ignoradas, os estados das etapas, os temas e a saída headless em PNG e SVG.

Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcWizard.WinForms.Community. Dependem dos pacotes Community do sgcUICore.	O mesmo conteúdo da edição Registered. Só a licença é diferente.

A edição Community é gratuita para uso comercial e não comercial por indivíduos e organizações com menos de EUR 1.000.000 de receita bruta anual e no máximo cinco desenvolvedores. Uma organização que atinja ou ultrapasse qualquer um dos limites precisa de uma licença Registered. O port .NET não tem edição Trial.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	VCL (TsgcVCLWizard)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	VCL e FMX	Windows, .NET Framework 4.7.2 e .NET 8 e 9. As páginas são declaradas em XAML.
Avalonia	FMX (TsgcFMXWizard)	Windows, Linux e macOS, .NET 8 e 9. As páginas são declaradas em XAML.
Headless	sgcWizard_Render	Qualquer plataforma dos frameworks de destino: SVG com esegece.sgcWizard, PNG com esegece.sgcWizard.Rendering.

Todos os hosts têm a mesma API: as páginas, ActivePageIndex, Next, Back, GoToPage, Finish e Cancel, os eventos OnBeforePageChange, OnPageChanged, OnFinish e OnCancel, os textos dos botões, StepperPosition, ShowCancel, Theme e AccentColor. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja Também

- Instalação via NuGet
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Renderização Headless no .NET
- Demos do .NET
- Plataformas

Instalação via NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, e os pacotes core e de renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcWizard.WinForms
dotnet add package esegece.sgcWizard.Wpf
dotnet add package esegece.sgcWizard.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcWizard.WinForms.Community
dotnet add package esegece.sgcWizard.Wpf.Community
dotnet add package esegece.sgcWizard.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcWizard.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie os pacotes Registered ou os pacotes Community em um projeto, nunca ambos: eles contêm os mesmos tipos no mesmo namespace.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:sgc="clr-namespace:esegece.sgcWizard;assembly=esegece.sgcWizard.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicação desktop Avalonia. Mapeie o namespace no XAML com `xmlns:sgc="using:esegece.sgcWizard"`.
- Headless: esegece.sgcWizard sozinho basta para SVG. Adicione esegece.sgcWizard.Rendering para PNG, ele traz o SkiaSharp e seus assets nativos para Windows e Linux.

Em C# todo host precisa de `using esegece.sgcWizard;`, e de `using esegece.sgcUICore;` para **TsgcUITheme** e os demais tipos dos temas.

Veja Também

- Visão Geral do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Demos do .NET

Início Rápido WinForms

TsgcWinFormsWizard é o assistente do WinForms, o equivalente do TsgcVCLWizard do Delphi. É um contêiner de páginas: cada **TsgcWinFormsWizardPage** é uma etapa, a faixa do stepper mostra as etapas e a barra de botões navega entre as páginas. O pacote é **esegece.sgcWizard.WinForms**.

Passos

- Crie um projeto WinForms e adicione o pacote esegece.sgcWizard.WinForms.
- Crie um **TsgcWinFormsWizard**, encaixe-o no formulário e adicione uma página por etapa com **AddPage**. Defina a **Description** de cada página: é a descrição da sua etapa, e o texto (caption) é o argumento de AddPage.
- Adicione os controles de cada página em **Controls** da página.
- Atribua **OnBeforePageChange** para validar uma página e defina aAllow como false quando ela não estiver completa, e **OnFinish** para usar os dados.
- Execute: Next e Enter avançam, Back volta, Finish aparece na última página, e Cancel ou Escape chamam OnCancel.

Exemplo

O Program.cs completo de uma aplicação WinForms com um assistente de três páginas. Ele compila como está em um projeto `net8.0-windows` que referencia esegece.sgcWizard.WinForms.
As páginas também podem ser criadas com `new TsgcWinFormsWizardPage()` e adicionadas com `Controls.Add`: a ordem das páginas é a ordem em que foram adicionadas. O assistente e o stepper estão na caixa de ferramentas, mas não há o verbo Add Page, veja Diferenças do .NET em relação ao Delphi.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WPF
- Início Rápido Avalonia
- Validação

Início Rápido WPF

TsgcWpfWizard é o assistente do WPF. Suas páginas são os elementos **TsgcWpfWizardPage** declarados em XAML como filhos dele, e cada página contém um elemento de conteúdo, normalmente um painel. O pacote é **esegece.sgcWizard.Wpf**.

Passos

- Crie um projeto WPF (`dotnet new wpf`) e adicione o pacote esegece.sgcWizard.Wpf.
- Mapeie o namespace `esegece.sgcWizard` e declare um **TsgcWpfWizard** com um **TsgcWpfWizardPage** por etapa. **Caption** e **Description** de cada página são o texto e a descrição da sua etapa.
- Atribua os eventos no code behind: são delegates, não eventos do WPF, portanto não podem ser definidos como atributos XAML.

Exemplo

O MainWindow.xaml declara o assistente e suas três páginas. O App.xaml do projeto é o do template do WPF, com StartupUri definido como MainWindow.xaml.
O MainWindow.xaml.cs atribui os eventos e valida a página Account.
As páginas também podem ser criadas em código com `Wizard.AddPage("Caption")`, que retorna o novo **TsgcWpfWizardPage**, e o assistente as expõe em **Pages**.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WinForms
- Início Rápido Avalonia
- Validação

Início Rápido Avalonia

TsgcAvaloniaWizard é o assistente do Avalonia, o equivalente do TsgcFMXWizard do Delphi, no Windows, Linux e macOS. Suas páginas são os elementos **TsgcAvaloniaWizardPage** declarados em XAML como filhos dele, e cada página contém um filho. O pacote é **esegece.sgcWizard.Avalonia**.

Passos

- Crie um projeto .NET 8 e adicione os pacotes esegece.sgcWizard.Avalonia, Avalonia.Desktop e Avalonia.Themes.Fluent.
- Mapeie o namespace com `xmlns:sgc="using:esegece.sgcWizard"` e declare um **TsgcAvaloniaWizard** com um **TsgcAvaloniaWizardPage** por etapa, com seus **Caption** e **Description**.
- Atribua os eventos no code behind: são delegates, não eventos do Avalonia.

Exemplo

O MainWindow.axaml declara o assistente e suas três páginas.
O MainWindow.axaml.cs define o tema, atribui os eventos e mostra uma mensagem na página Account enquanto o nome estiver vazio.
O Program.cs inicia a aplicação com o tema Fluent.
As páginas também podem ser criadas em código com `Wizard.AddPage("Caption")`, que retorna o novo **TsgcAvaloniaWizardPage**; defina seu **Child** como o conteúdo da página.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WinForms
- Início Rápido WPF
- Validação

Classes do .NET

Todo host tem um assistente, uma página, um stepper e uma barra de botões com os mesmos membros. Os tipos independentes de framework são compartilhados pelos três hosts e mantêm seus nomes do Delphi.

Classes dos hosts

Função	WinForms	WPF	Avalonia
Assistente	TsgcWinFormsWizard	TsgcWpfWizard	TsgcAvaloniaWizard
Página	TsgcWinFormsWizardPage (um contêiner de Control)	TsgcWpfWizardPage (um elemento Content)	TsgcAvaloniaWizardPage (um Border com um Child)
Coleção de páginas	ReadOnlyCollection<TsgcWinFormsWizardPage>	TsgcWpfWizardPages	AvaloniaList<TsgcAvaloniaWizardPage>
Stepper	TsgcWinFormsStepper	TsgcWpfStepper	TsgcAvaloniaStepper
Barra de botões	TsgcWinFormsWizardButtonBar	TsgcWpfWizardButtonBar	TsgcAvaloniaWizardButtonBar

Tipos independentes de framework

Tipo	Descrição
TsgcStepperModel	As etapas, a etapa ativa, a orientação, os estados automáticos, o layout e o hit test. O stepper de cada host possui um, veja TsgcStepperModel.
TsgcWizardSteps, TsgcWizardStep	A coleção de etapas e uma etapa: Caption, Description, Glyph, State, Enabled, Visible e 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	O renderizador e a saída headless em SVG e PNG, veja Renderização Headless no .NET.

Assistente

Membro	Tipo	Descrição
Pages	coleção	As páginas em ordem. PageCount e IndexOfPage(aPage) como no Delphi.
AddPage(string aCaption)	página	Cria uma página, adiciona-a no final e a retorna.
ActivePageIndex, ActivePage	int, página	A página visível. Defini-los troca a página diretamente, sem eventos.
Next(), Back(), GoToPage(int aIndex)	bool	Vai para outra página através de OnBeforePageChange, ignorando as páginas cujo Skip é true. Next na última página chama Finish.
Finish(), Cancel()	bool, void	Finish chama OnBeforePageChange com aToIndex = PageCount e depois OnFinish. Cancel chama OnCancel.
NextPageIndex(int), PrevPageIndex(int), IsLastPage()	int, bool	A próxima página e a página anterior que não são ignoradas, e se a página ativa é a última.
StepperPosition	TsgcWizardStepperPosition	Onde fica a faixa do stepper: em cima, à esquerda ou oculta.
BackCaption, NextCaption, FinishCaption, CancelCaption	string	Os textos dos botões, para localização.
ShowCancel	bool	Mostra o botão Cancel.
Theme, AccentColor	TsgcUITheme, uint	O tema do assistente, da sua faixa do stepper, da sua barra de botões e dos controles das suas páginas. AccentColor 0 usa o destaque do tema.
Stepper, ButtonBar	stepper, barra de botões	A faixa do stepper e a barra de botões internas.
OnBeforePageChange	TsgcWizardPageChangeEvent	Antes de cada mudança de página e antes de Finish. aAllow false a impede.
OnPageChanged, OnFinish, OnCancel	TsgcUINotifyEvent	Depois de uma mudança de página, em Finish e em Cancel.

Página

Membro	Tipo	Descrição
Caption	string	O texto (caption) da etapa. No WinForms é o Text do controle.
Descrição	string	A descrição da etapa e a sua dica (tooltip).
StepGlyph	string	O texto dentro do círculo da etapa no lugar do seu número.
Skip	bool	Next, Back e GoToPage pulam a página e a etapa é desenhada como ignorada.
Wizard, PageIndex	assistente, int	O assistente da página e o seu índice.
Content (WPF), Child (Avalonia), Controls (WinForms)	elemento	O que a página mostra.

Stepper

O stepper funciona sozinho, sem um assistente. Seus membros são os dos steppers do Delphi: Model, Steps, Orientation, ActiveIndex, AutoStates, ShowDescriptions, CircleSize, ConnectorThickness, Theme, AccentColor, Clickable, StepAt(X, Y), StepRect(aIndex), OnStepClick (TsgcWizardStepClickEvent, aAllow false impede o clique) e OnActiveChange, que dispara apenas por interação do usuário. Wizard é o assistente que o possui, ou null para um stepper independente. Veja TsgcVCLStepper.

Barra de botões

A barra de botões de um assistente desenha Back, Next (Finish na última página) e Cancel. GetButtons(), ButtonCount(), Buttons(aIndex), ButtonRect(aIndex) e ButtonAt(X, Y) descrevem os botões, e Wizard é o seu assistente. Ela é criada pelo assistente e acessada pela sua propriedade ButtonBar.
Posições e tamanhos são pixels no WinForms (Rectangle) e unidades independentes de dispositivo no WPF e no Avalonia (Rect).

Veja Também

- Visão Geral do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia

Diferenças do .NET em relação ao Delphi

A API .NET segue a API Delphi nome por nome. Estes são os pontos em que ela difere.

Eventos

OnBeforePageChange, OnPageChanged, OnFinish, OnCancel, OnStepClick e OnActiveChange são propriedades delegate com os nomes do Delphi, não membros `event` do C#. Atribuir um handler substitui o anterior, como um ponteiro de método do Delphi, e eles são atribuídos em código, também no WPF e no Avalonia, onde o XAML não pode defini-los. O parâmetro `var aAllow: Boolean` do Delphi é `ref bool aAllow`, e um handler de OnBeforePageChange tem a assinatura `void (object Sender, int aFromIndex, int aToIndex, ref bool aAllow)`. Os inícios rápidos mostram os handlers dos três hosts.

GoToPage e navegação

GoToPage, Next, Back e Finish retornam bool e se comportam como no Delphi: GoToPage retorna false quando o índice está fora do intervalo ou a página é ignorada, retorna true quando a página já está ativa, e caso contrário passa por OnBeforePageChange. Finish chama OnBeforePageChange com aToIndex = PageCount, de modo que aToIndex > aFromIndex vale para todo avanço. Um clique em uma etapa concluída da faixa do stepper chama GoToPage. Definir ActivePageIndex ou ActivePage troca a página sem eventos, como no Delphi.

A coleção de páginas

O Delphi tem a propriedade indexada `Pages[Index]` e `PageCount`. No .NET **Pages** é uma coleção, então `Pages[i]`, `Pages.Count` e `foreach` funcionam como de costume:

- WinForms: uma coleção somente leitura. Uma página é adicionada com AddPage ou adicionando um TsgcWinFormsWizardPage aos Controls do assistente, e a ordem das páginas é a ordem em que foram adicionadas, não a ordem z dos controles.
- WPF: TsgcWpfWizardPages, a propriedade de conteúdo do assistente, portanto as páginas são os filhos do assistente no XAML. Cada página tem um elemento Content.
- Avalonia: uma AvaloniaList, a propriedade de conteúdo do assistente. Cada página é um Border com um Child.

Dicas e tooltips

No Delphi a dica de uma etapa é a sua Description, exibida quando ShowHint é True, e o FMX mostra dicas a partir do 10.1 Berlin. No .NET o stepper sempre mostra a Description da etapa sob o mouse como um tooltip, sem propriedade ShowHint: o WinForms usa um componente ToolTip interno, o WPF define a propriedade ToolTip do stepper e o Avalonia define a propriedade anexada ToolTip.Tip. Não defina você mesmo o tooltip de um stepper, o stepper o substitui; defina a Description da etapa ou da página.

Sem verbos do designer

O editor de componente do Delphi tem os verbos Add Page, Next Page e Previous Page e mostra outra página no designer quando uma etapa é clicada, veja Tempo de Design. O port .NET não tem verbos do designer. No WinForms o assistente e o stepper estão na caixa de ferramentas, e as páginas são criadas em código com AddPage. No WPF e no Avalonia as páginas são declaradas em XAML.

Teclado

Enter é Next ou Finish e Escape é Cancel enquanto o foco está dentro do assistente, como no Delphi. No WinForms as teclas chegam por ProcessDialogKey, então um TextBox multilinha ou um botão com foco as mantém.

No WPF e no Avalonia elas chegam pelo evento KeyDown em bubbling, então um controle que trata a tecla prevalece.

Edições

O port .NET é distribuído em pacotes Registered e Community com o mesmo conteúdo, veja Visão Geral do .NET. Não há edição Trial nem chave de compilação no código.

Vinculação de dados

No WPF e no Avalonia, as propriedades visíveis ao usuário são propriedades de dependência e propriedades do Avalonia, portanto podem ser vinculadas, estilizadas e animadas, veja Vinculação de dados no .NET.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia

Vinculação de dados no .NET

Nos hosts WPF e Avalonia, as propriedades visíveis ao usuário do assistente, de suas páginas e do stepper são propriedades de dependência (WPF) e propriedades do Avalonia, portanto podem ser vinculadas a um view model, definidas por um Style e animadas. Este tópico se aplica somente a WPF e Avalonia. Os controles WinForms não mudam: suas propriedades são propriedades CLR simples, como antes.

Propriedades vinculáveis

Cada propriedade da tabela tem um campo estático com o sufixo Property, por exemplo `TsgcWpfWizard.ActivePageIndexProperty`. No WPF os campos são DependencyProperty. No Avalonia são StyledProperty, exceto ActivePageIndex e ActiveIndex, que são DirectProperty. As propriedades marcadas com **TwoWay** são vinculadas em TwoWay por padrão, porque o próprio controle as altera (Next, Back, GoToPage, um clique em uma etapa), de modo que uma vinculação sem Mode mantém o view model atualizado. As demais propriedades são vinculadas em OneWay por padrão.

Classe WPF	Classe Avalonia	Propriedades vinculáveis
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

No Avalonia, Theme é o tema do sgcUICore (TsgcUITheme) e oculta o tema de controle do Avalonia de StyledElement, que continua acessível por meio de `StyledElement.ThemeProperty`.

Comportamento

- **Uma vinculação navega, o código não.** Um novo ActivePageIndex que chega por uma vinculação de dados é uma navegação: passa por OnBeforePageChange e depois por OnPageChanged, como GoToPage. Definido pelo código (o setter CLR), ActivePageIndex troca a página sem eventos, como no Delphi.
- **Um veto é gravado de volta.** Quando OnBeforePageChange define aAllow como false, o assistente mantém a página atual e restaura o índice antigo no controle e no view model vinculado. No WPF, o view model recebe o índice antigo depois que o dispatcher executa o trabalho pendente.
- **Um índice fora do intervalo é ignorado.** Um índice vinculado que não é uma página mantém a página atual e não dispara nenhum evento, e uma origem TwoWay recebe de volta o índice em uso, no WPF e no Avalonia. A vinculação continua ativa, de modo que o próximo índice válido navega.
- **Um valor corrigido chega à origem.** Um valor que o controle corrige, como um CircleSize ou um ActiveIndex que o stepper limita, é gravado de volta na propriedade e em uma origem TwoWay, no WPF e no Avalonia. No WPF a origem o recebe depois que o dispatcher executa o trabalho pendente. No Avalonia a gravação de volta cobre um valor que a origem envia quando a vinculação já está em execução. Um valor que a origem já tem quando a vinculação inicia é corrigido no controle, mas não é gravado de volta na origem.
- **O view model acompanha o assistente.** Next, Back, GoToPage, um clique em uma etapa e uma alteração pelo código copiam o novo índice para ActivePageIndex, e uma vinculação TwoWay o repassa ao view model.
- **Styles.** Um setter de Style pode definir qualquer propriedade vinculável, por exemplo o Theme e as opções do stepper de todos os assistentes de uma janela.

Propriedades que continuam sendo propriedades CLR

Propriedade	Motivo
Pages (assistente)	A coleção de páginas é a propriedade de conteúdo do assistente, preenchida pelas páginas declaradas em XAML, e não um valor para vincular.
ActivePage (assistente)	Uma referência a um elemento de página. Vincule ActivePageIndex em vez disso, que acompanha cada troca de página.
PageIndex (página)	Somente leitura, a posição da página em Pages.
Steps (stepper)	A coleção de etapas. O assistente preenche as etapas do seu próprio stepper a partir das páginas.
OnBeforePageChange, OnPageChanged, OnFinish, OnCancel, OnStepClick, OnActiveChange	Propriedades delegate com os nomes do Delphi, atribuídas no código, veja Diferenças do .NET em relação ao Delphi.

Exemplo

O view model abaixo é compartilhado pelos dois exemplos. PageIndex está vinculado a ActivePageIndex sem Mode, portanto é vinculado em TwoWay, NextCaption está vinculado ao título do botão Next, e SkipOptions está vinculado à propriedade Skip da página Options e a uma caixa de seleção.

WPF

MainWindow.xaml cria o view model como DataContext da janela. O code-behind e o App.xaml são os do modelo WPF.

Avalonia

MainWindow.axaml define `x:DataType` como o view model, de modo que as vinculações são vinculações compiladas. O code-behind chama InitializeComponent, e a classe App é a de Início Rápido Avalonia.

Veja Também

- Diferenças do .NET em relação ao Delphi
- Classes do .NET
- Início Rápido WPF
- Início Rápido Avalonia

Renderização Headless no .NET

Um stepper pode ser desenhado sem janela, em um servidor, em um teste ou em um relatório. **sgcWizard_Render.sgcWizardSaveToSVG** grava um arquivo ou stream SVG apenas com o pacote esegece.sgcWizard, e **sgcWizard_Render_Raster.sgcWizardSaveToPNG** grava um PNG com esegece.sgcWizard.Rendering, que desenha no SkiaSharp no Windows, Linux e macOS. A saída SVG é idêntica byte a byte à da renderização headless do Delphi.

Métodos

Método	Pacote	Descrição
sgcWizard_Render.sgcWizardSaveToSVG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	esegece.sgcWizard	Grava o stepper de aModel como SVG.
sgcWizard_Render_Raster.sgcWizardSaveToPNG(aModel, aFileName or aStream, aWidth, aHeight, aScale = 1)	esegece.sgcWizard.Rendering	Grava o stepper de aModel como PNG sobre fundo transparente.
sgcWizard_Render.sgcWizardSaveButtonBarToSVG(aButtons, aStream, aWidth, aHeight, aScale, aTheme, aAccent)	esegece.sgcWizard	Grava uma barra de botões como SVG.
sgcWizard_Render_Raster.sgcWizardSaveButtonBarToPNG(aButtons, aStream, aWidth, aHeight, aScale, aTheme, aAccent)	esegece.sgcWizard.Rendering	Grava uma barra de botões como PNG.
sgcWizard_Render.sgcWizardButton(aCaption, aEnabled = true, aIsDefault = false, aHot = false, aPressed = false)	esegece.sgcWizard	Retorna um TsgcWizardButton para uma barra de botões.

sgcWizard_Render.sgcWizardRender(aModel, aCanvas, aWidth, aHeight, aScale = 1)	esegece.sgcWizard	Desenha o stepper em qualquer canvas do sgcUICore.

aWidth e aHeight são pixels lógicos. A imagem tem aWidth x aScale por aHeight x aScale pixels, então uma escala de 2 gera uma imagem nítida para uma tela de alta DPI ou para impressão.

Exemplo

Um programa de console (`net8.0`) que referencia esegece.sgcWizard.Rendering e grava um stepper escuro com uma etapa ignorada como PNG e SVG, as mesmas etapas desenhadas na vertical e uma barra de botões.

Veja Também

- Visão Geral do .NET
- Classes do .NET
- Renderização headless
- TsgcStepperModel

Demos do .NET

As demos do .NET estão na pasta `demos` do produto e no zip de demos, com a solução `sgcWizard_Demos.sln`. Cada demo tem um README.md.

Demo	Host	Descrição
Onboarding	WinForms	Um assistente de conta de quatro páginas (Welcome, Account, Options e Summary) em um TsgcWinFormsWizard. OnBeforePageChange valida o nome e o e-mail da página Account, uma caixa de seleção define Skip na página Options, OnPageChanged preenche o resumo, OnFinish o exibe e OnCancel pergunta antes de fechar. A barra de ferramentas alterna o tema e coloca a faixa do stepper em cima, à esquerda ou a oculta.
OnboardingWpf	WPF	O mesmo assistente em um TsgcWpfWizard declarado em MainWindow.xaml.
OnboardingAvalonia	Avalonia	O mesmo assistente em um TsgcAvaloniaWizard, no Windows, Linux e macOS.
Stepper	WinForms	Dois controles TsgcWinFormsStepper independentes, um horizontal e outro vertical, com as cinco etapas de um checkout, mantidos em sincronia. Os botões avançam, voltam, marcam a etapa ativa como erro, a ignoram e reiniciam as etapas, e um clique em uma etapa concluída a ativa através de OnStepClick.
StepperWpf	WPF	Os mesmos steppers como controles TsgcWpfStepper colocados em MainWindow.xaml.
StepperAvalonia	Avalonia	Os mesmos steppers como controles TsgcAvaloniaStepper.

Compilar e executar

As demos precisam do SDK do .NET 8 ou posterior. Execute uma demo a partir da sua pasta:

```
dotnet run
```

Dentro da árvore do produto as demos compilam contra os projetos da biblioteca. A partir do zip de demos elas restauram os pacotes NuGet, a edição Community por padrão. Defina **SgcWizardEdition** para usar os pacotes Registered:

```
dotnet build -p:SgcWizardEdition=Registered
```

Autoverificação

Toda demo tem uma opção `--test` que abre a janela fora da tela, aciona o assistente ou os steppers, grava uma linha PASS ou FAIL por verificação no arquivo e termina com o número de verificações que falharam, então 0 significa que todas passaram.

```
dotnet run -- --test result.txt
```

Veja Também

- Visão Geral do .NET
- Instalação via NuGet

- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Demos

Solução de Problemas

EsgcWizardError: A TsgcVCLWizardPage can only be placed on a TsgcVCLWizard

Uma página deve ser filha direta de um assistente. Definir o Parent de um TsgcVCLWizardPage como um painel, um formulário ou qualquer outro controle gera EsgcWizardError, e o mesmo vale para um TsgcFMXWizardPage fora de um TsgcFMXWizard. Crie as páginas com **AddPage** ou defina o Parent delas como o assistente. Para agrupar controles em uma página, coloque um TPanel ou um frame na página. Consulte TsgcVCLWizardPage.

Um programa de teste solicita direitos de administrador

O Windows trata um executável cujo nome contém palavras como setup, install ou update como um instalador e solicita elevação antes de iniciá-lo. Ao testar um assistente, não nomeie o programa como SetupWizard.exe ou InstallWizard.exe. Escolha um nome como Onboarding.exe ou adicione um manifesto com requestedExecutionLevel asInvoker.

Next não faz nada

OnBeforePageChange definiu aAllow como False, que é a forma como a validação veta uma mudança, ou as próximas páginas estão todas ignoradas, de modo que Next chama Finish. Verifique o manipulador e a propriedade Skip das páginas. Consulte Validação.

Finish não fecha o formulário

O assistente nunca fecha nada. Feche o formulário ou defina ModalResult em OnFinish e em OnCancel. Consulte Finish e Cancel.

Enter não avança para a próxima página

Um controle que quer o Enter fica com ele: um TMemo, um botão com foco ou um TButton no formulário com Default definido como True. Consulte Teclado.

Um clique em uma etapa não muda a página

Somente etapas concluídas, e etapas com erro anteriores à ativa, podem ser clicadas, e apenas enquanto Clickable for True. Um clique também passa por OnBeforePageChange.

Um edit FMX mantém suas cores claras em uma página escura

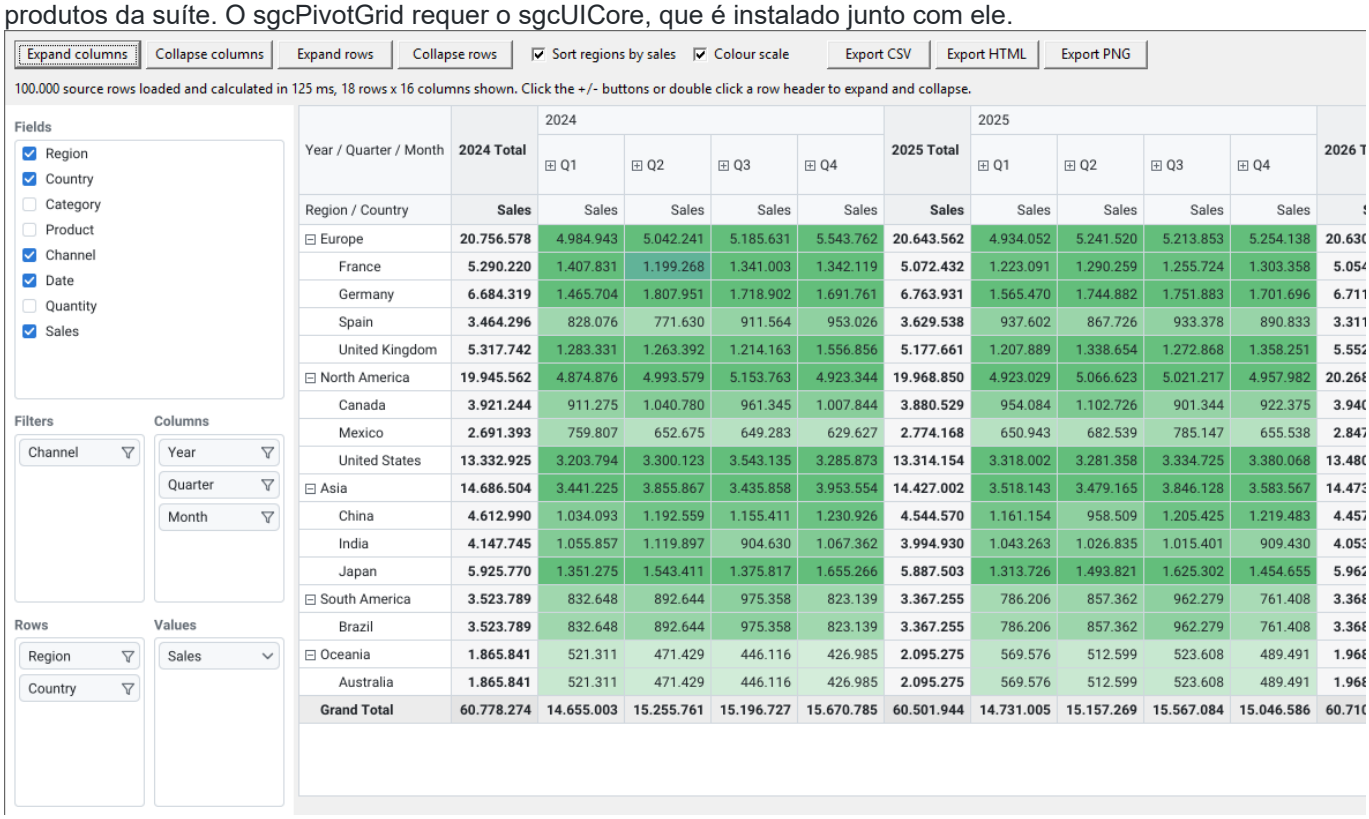
Os controles edit estilizados do FireMonkey mantêm as cores do seu estilo. Use um estilo escuro para o formulário ou defina as cores do controle. Consulte Temas.

Os controles FMX não aparecem

Os controles FireMonkey exigem o Delphi XE6 ou posterior, e as dicas dos controles FMX exigem o 10.1 Berlin ou posterior. Consulte Plataformas.

Introdução

sgcPivotGrid adiciona tabelas dinâmicas a aplicativos Delphi e C++Builder, para VCL e FireMonkey. **TsgcVCLPivotGrid** e **TsgcFMXPivotGrid** resumem linhas de origem planas pelos campos que você coloca nas linhas, nas colunas, nos filtros e nos valores: somas, contagens, médias e mais, com subtotais e totais gerais, agrupamento de datas por ano, trimestre, mês, semana ou dia, filtros, ordenação, expandir e recolher, drill-down até as linhas de origem e exportação para CSV, JSON, HTML, PNG e SVG. **TsgcVCLPivotFieldList** e **TsgcFMXPivotFieldList** permitem que o usuário monte o layout arrastando e soltando.
Os dados vêm de um **TDataSet** por meio de **TsgcPivotSource_DataSet**, de qualquer **IsgcUITable**, como um sgcGrid, por meio de **TsgcPivotSource_Table**, ou do seu próprio código com AppendRow. O cálculo é feito por **TsgcPivotEngine**, uma classe não visual com um armazenamento de colunas tipado, de modo que a mesma tabela dinâmica também roda sem janela, em um servidor ou no Linux64, e **TsgcPivotGridRenderer** a desenha em PNG e SVG. Um milhão de linhas de origem são recalculadas em uma fração de segundo.
O sgcPivotGrid faz parte da suíte visual da eSeGeCe. Ele é desenhado com **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro, de modo que a tabela dinâmica tem a mesma aparência na VCL, no FireMonkey e na saída headless, e combina com os outros produtos da suíte. O sgcPivotGrid requer o sgcUICore, que é instalado junto com ele.



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

Edições

O sgcPivotGrid é distribuído nas edições Registered, Trial e Community. A edição Community deixa de fora quatro recursos avançados: campos calculados, mostrar valores como, Top N e persistência do layout. Veja Edições.

Por onde começar

- Arquitetura: as origens, o armazenamento de colunas, o pipeline do engine, o snapshot e as visões.
- Plataformas: as versões do Delphi, o C++Builder e as plataformas.
- Início rápido VCL, Início rápido FMX, Início rápido C++Builder, Início rápido DataSet e Início rápido headless: uma primeira tabela dinâmica em poucos minutos.
- Componentes: as grades, as listas de campos, o engine e seus campos, as origens, o renderer e as regras de formatação.
- Áreas, Agregações, Agrupamento de datas, Totais, Filtros, Top N, Mostrar valores como, Campos calculados, Persistência do layout e Exportação.

- Expandir e recolher, Ordenação, Drill-down, Formatação condicional, Interoperabilidade com gráficos, Teclado e acessibilidade e Desempenho.
- Demos: os projetos de demonstração em Delphi e C++Builder.
- Solução de problemas: problemas comuns e suas causas.

Arquitetura

O sgcPivotGrid mantém o cálculo separado do desenho, e o desenho separado do framework. O engine e o renderer usam apenas a RTL e o sgcUICore, e os controles VCL e FMX são hosts finos em torno deles.

Fluxo de dados

1. **Origens.** TsgcPivotSource_DataSet lê um TDataSet, TsgcPivotSource_Table lê qualquer IsgcUITable, e seu código pode chamar AppendRow de TsgcPivotEngine diretamente. Todas gravam no mesmo armazenamento de colunas.
2. **Armazenamento de colunas.** As colunas de texto são codificadas por dicionário. Números, datas e booleanos são valores Double, com NaN como valor vazio. O tipo de uma coluna vem do tipo do campo DB, de DefineColumn ou de seu primeiro valor que não seja vazio, e uma coluna inteira se torna uma coluna de ponto flutuante quando chega um valor fracionário.
3. **Engine.** Cada recálculo codifica os membros dos campos de linha, coluna e filtro, aplica os filtros, agrega as células folha em uma única passagem sobre as linhas de origem, aplica os filtros de valor e o Top N, acumula as folhas nos subtotais e totais gerais com mesclagens exatas, ordena os membros e achata os eixos com seu estado de expansão. Os totais são calculados a partir dos valores de origem, nunca como médias de médias.
4. **Snapshot.** O resultado é publicado como um TsgcPivotResult imutável. Os leitores o obtêm entre Lock e Unlock, de modo que uma visão sempre desenha um resultado completo. Expandir ou recolher um membro constrói um novo snapshot sobre os mesmos dados, sem agregar novamente.
5. **Saída.** TsgcPivotGridRenderer desenha um snapshot em qualquer canvas do sgcUICore: a tela, um PNG ou um SVG. TsgcVCLPivotGrid e TsgcFMXPivotGrid hospedam o renderer com rolagem, seleção, teclado e mouse. Os métodos de exportação gravam a visão como CSV, HTML ou JSON, e ResultTable a expõe como uma IsgcUITable, por exemplo para o sgcCharts.

Classes

Classe	Unidade	Função
TsgcVCLPivotGrid	sgcPivotGrid_VCL	A grade dinâmica VCL. Possui um engine, publicado como Pivot.
TsgcFMXPivotGrid	sgcPivotGrid_FMX	A grade dinâmica FireMonkey.
TsgcVCLPivotFieldList	sgcPivotGrid_FieldList_VCL	A lista de campos VCL, vinculada a uma grade ou a um engine.
TsgcFMXPivotFieldList	sgcPivotGrid_FieldList_FMX	A lista de campos FireMonkey.
TsgcPivotEngine	sgcPivotGrid_Engine	O armazenamento de colunas, o layout, o cálculo, o snapshot, o drill-down e a exportação.
TsgcPivotField	sgcPivotGrid_Fields	Um campo do layout: sua área, agrupamento, agregação, ordenação e filtros.
TsgcPivotCalculatedField	sgcPivotGrid_Fields	Uma coluna calculada por linha de origem a partir de uma expressão.
TsgcPivotSource_DataSet	sgcPivotGrid_Source_DataSet	Carrega um TDataSet e o recarrega quando os dados mudam.
TsgcPivotSource_Table	sgcPivotGrid_Source_Table	Carrega qualquer IsgcUITable e a recarrega quando sua versão muda.

TsgcPivotGridRenderer	sgcPivotGrid_Render	Desenha um snapshot em um canvas, PNG ou SVG, e faz o teste de acerto.
TsgcPivotFormatRule	sgcPivotGrid_Format	Uma regra de formatação condicional: uma regra de valor ou uma escala de cores.

Threads

Um recálculo roda na thread que o dispara, normalmente a thread principal, e OnCalculated é gerado nessa thread. O snapshot é protegido por um bloqueio, portanto o código em outra thread lê Result somente entre Lock e Unlock. TsgcPivotSource_Table recarrega na thread que notifica a alteração de sua tabela.

Desempenho

O grupo de desempenho do QA mede esses cenários com um build de release: uma execução de aquecimento e depois a mediana de cinco execuções, com dados de uma semente fixa para que cada execução seja idêntica. A tabela lista o orçamento de cada cenário.

Cenário	Win64	Win32
Recalcular 1.000.000 de linhas, 2 campos de linha (50 e 20 membros), 1 campo de coluna (12 membros), 1 Sum	100 ms	160 ms
AppendRow de 1.000.000 de linhas com 3 valores de texto, 1 float e 1 data	900 ms	1,400 ms
O primeiro layout com CountDistinct em uma coluna de 100.000 valores	400 ms	600 ms
O primeiro layout com StdDev, Average e Max como três medidas	180 ms	280 ms
Um quadro em 1920x1080 de 10.000 linhas e 50 colunas de dados, rolado a cada quadro	16 ms	25 ms
Expandir ou recolher um membro de um eixo com 10.000 itens	5 ms	8 ms
Memória do engine acima do armazenamento no primeiro cenário	32 MB	32 MB

Veja também

- Plataformas
- TsgcPivotEngine
- Início rápido headless

Edições

O sgcPivotGrid é compilado a partir do mesmo código-fonte para todas as edições.

Edição	Inclui
Registered	Tudo: as grades e listas de campos VCL e FMX, o engine, as duas origens, todas as agregações, agrupamento de datas, totais, filtros, ordenação, drill-down, regras de formatação, exportação, a saída headless em PNG e SVG, e os quatro recursos avançados: campos calculados, mostrar valores como, Top N e persistência do layout.
Trial	Os mesmos recursos da Registered. Um build compilado com SGC_TRIAL mostra uma mensagem de que é uma versão de avaliação quando um componente é criado depois que a janela de lançamento indicada no build passou. Nada mais muda.
Community	Tudo, exceto os quatro recursos avançados. CountDistinct, StdDev, Var, agrupamento de datas, filtros, drill-down, exportação, regras de formatação, as listas de campos e as visões fazem parte da Community, e não há aviso de avaliação.

Os recursos avançados

Os recursos avançados são compilados sob o define SGC_PIVOTGRID_ADVANCED, que o sgcVer.inc desativa na edição Community. Suas units, sgcPivotGrid_Expr, sgcPivotGrid_ShowAs e sgcPivotGrid_Layout, não fazem parte da distribuição Community, e estes membros não existem em um build Community:

Recurso	Membros
Campos calculados	TsgcPivotEngine.CalculatedFields, TsgcPivotCalculatedField, TsgcPivotCalculatedFields
Mostrar valores como	TsgcPivotField.ShowAs, ShowAsAxis e ShowAsBaseItem
Top N	TsgcPivotField.TopN e TopNOthers, TsgcPivotEngine.OthersCaption
Persistência do layout	SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile e LoadLayoutFromFile do engine e das duas grades

A edição é uma escolha em tempo de compilação no sgcVer.inc. SGC_COMMUNITY também desativa SGC_TRIAL. O código que usa um membro avançado e também precisa compilar na Community o coloca dentro de {$IFDEF SGC_PIVOTGRID_ADVANCED}. Um formulário salvo por um build Registered com propriedades avançadas definidas não carrega em um build Community.

Veja também

- Plataformas
- Introdução

Instalação

O sgcPivotGrid faz parte da distribuição sgcUI. Um único instalador cobre os produtos sgcUI e instala apenas os produtos que sua licença inclui.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e faça login com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcPivotGrid** e quaisquer outros produtos desejados. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que sua licença não inclui ficam esmaecidos. Uma licença All-Access libera todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, em ordem de dependência, e instala os pacotes de tempo de design.

O que acompanha o sgcPivotGrid

O **sgcUICore**, a base compartilhada dos produtos visuais do sgcUI, acompanha o sgcPivotGrid. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcPivotGrid é selecionado.

Pastas instaladas

Cada produto é instalado em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para a versão de avaliação e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de todos os produtos instalados. Este é o único caminho de biblioteca do sgcUI.
`Packages`	Os pacotes de todos os produtos instalados, um conjunto por produto e versão do IDE.
`Lib<V>`	As units compiladas para cada versão do IDE e plataforma, por exemplo `LibD13` .
`Demos\PivotGrid`	As demos do sgcPivotGrid. Veja Demos.

O instalador adiciona `sgcUI\Source` ao caminho de biblioteca e ao caminho de navegação de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, portanto não precisa de um caminho próprio.

Trial e Community

- **Trial**: o instalador de avaliação do sgcUI instala todos os produtos em modo de avaliação, sgcPivotGrid inclusive.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcPivotGrid sem seus quatro recursos avançados. Veja Edições.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão do IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo da versão do Delphi, por exemplo `sgcPivotGridD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcPivotGrid<V>` / `dclsgcPivotGrid<V>`	Os componentes do sgcPivotGrid (runtime / tempo de design).

Compile-os em ordem de dependência: `sgcUICore`, depois `sgcPivotGrid`, depois `dclsgcPivotGrid`. Em seguida instale o pacote de tempo de design `dclsgcPivotGrid` e adicione `sgcUI\Source` ao caminho de biblioteca da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcPivotGridC13.cbproj`, e compile-o. Os cabeçalhos `.hpp` gerados aparecem ao lado do pacote compilado.

Veja também

- Plataformas
- Início rápido VCL

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	TsgcVCLPivotGrid e TsgcVCLPivotFieldList no Windows, Win32 e Win64, todas as versões do Delphi.
FireMonkey	TsgcFMXPivotGrid e TsgcFMXPivotFieldList a partir do Delphi XE6 (SGC_PIVOTGRID_FMX). Usam apenas o FireMonkey e o sgcUICore, com suas próprias barras de rolagem e gesto de arrastar por toque.
Linux64	As units independentes de framework a partir do Delphi 10.2 (SGC_PIVOTGRID_LINUX): o engine, o armazenamento de colunas, as duas origens, a exportação, o JSON de layout, o renderer com a saída headless em PNG e SVG, e o modelo da lista de campos.
.NET	Um port em C# com a mesma API para WinForms, WPF e Avalonia, a saída PNG e SVG sem interface e a origem DataTable, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão geral do .NET.

Pacotes

O sgcPivotGrid traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcPivotGridD13.dpk para o Delphi 13 e sgcPivotGridC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcPivotGrid	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, dbrtl, sgcUICore
sgcPivotGridVCL	Runtime, VCL	sgcPivotGrid_VCL e sgcPivotGrid_FieldList_VCL (TsgcVCLPivotGrid, TsgcVCLPivotFieldList)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcPivotGrid
sgcPivotGridFMX	Runtime, FireMonkey, a partir do XE6	sgcPivotGrid_FMX e sgcPivotGrid_FieldList_FMX (TsgcFMXPivotGrid, TsgcFMXPivotFieldList)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcPivotGrid
dclsgcPivotGrid	Tempo de projeto, VCL	Registra TsgcPivotSource_DataSet, TsgcPivotSource_Table, TsgcVCLPivotGrid e TsgcVCLPivotFieldList na página **SGC PivotGrid** da paleta, com o editor de componente, a lista suspensa de FieldName e os editores de coleção.	designide, rtl, vcl, sgcPivotGrid, sgcPivotGridVCL
dclsgcPivotGridFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXPivotGrid e TsgcFMXPivotFieldList na página **SGC PivotGrid** da paleta, com o editor de componente deles.	designide, rtl, fmx, sgcPivotGrid, sgcPivotGridFMX, dclsgcPivotGrid

sgcPivotGridWS, dclsgcPivotGridWS	Runtime e tempo de projeto, ponte do sgcWebSockets	A ponte de dados em tempo real, independente de framework.	rtl, sgcPivotGrid, sgcWebSockets

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcPivotGridFMX exige o dclsgcPivotGrid, portanto instale primeiro o dclsgcPivotGrid.

Versões do IDE e plataformas de destino

Delphi	sgcPivotGrid	sgcPivotGridVCL	sgcPivotGridFMX	dclsgcPivotGrid, dclsgcPivotGridFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcPivotGrid, sgcPivotGridVCL	sgcPivotGridFMX	dclsgcPivotGrid, dclsgcPivotGridFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcPivotGrid)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcPivotGrid)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcPivotGridC13.cbproj, sgcPivotGridVCLC13.cbproj, sgcPivotGridFMXC13.cbproj, dclsgcPivotGridC13.cbproj e

dclsgcPivotGridFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXPivotGrid e TsgcFMXPivotFieldList) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcPivotGridVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcPivotGrid_FMX e sgcPivotGrid_FieldList_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcPivotGrid é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa o motor pivot e as fontes. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Restrições de versão

- Toda unit compila no Delphi 7: sem generics, sem métodos anônimos, sem variáveis inline e sem record helpers.
- Os cabeçalhos do C++Builder mantêm prefixos de enumeração exclusivos (pva, pvf, pvd, pvx...), de modo que nunca conflitam com o sgcGrid ou o sgcCharts.

Veja também

- Edições
- Arquitetura

Início rápido VCL

Instale os pacotes do sgcPivotGrid para a sua versão do Delphi. Os componentes estão na página de paleta **SGC PivotGrid**.

Em tempo de design

1. Solte um **TsgcVCLPivotGrid** em um formulário e defina Align como alClient.
2. Solte um **TsgcVCLPivotFieldList**, defina Align como alRight e defina sua propriedade PivotGrid como a grade.
3. Solte um **TsgcPivotSource_DataSet**, defina DataSet como seu dataset, PivotGrid como a grade e Active como True.
4. Expanda a propriedade Pivot da grade e adicione itens a Pivot.Fields. Escolha uma coluna na lista suspensa de FieldName e defina Area: pvaRow, pvaColumn, pvaFilter ou pvaData. Um campo de dados também tem uma Aggregation, Sum por padrão.
5. Execute o aplicativo. O usuário reorganiza os campos com a lista de campos, e a grade recalcula após cada alteração.

No código

A grade possui seu engine, a propriedade Pivot. Carregue as linhas entre BeginLoad e EndLoad e altere o layout entre BeginUpdate e EndUpdate, para que cada bloco recalcule uma 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;
```

Veja também

- TsgcVCLPivotGrid
- TsgcVCLPivotFieldList
- Início rápido DataSet
- Áreas

Início rápido FMX

Os componentes FireMonkey exigem o Delphi XE6 ou posterior. Eles estão na página de paleta **SGC PivotGrid** de um formulário FireMonkey.

Em tempo de design

1. Solte um **TsgcFMXPivotGrid** em um formulário e defina Align como Client.
2. Solte um **TsgcFMXPivotFieldList**, defina Align como Right e defina sua propriedade PivotGrid como a grade.
3. Vincule os dados com um **TsgcPivotSource_DataSet** ou um **TsgcPivotSource_Table**, ou carregue-os no código, e adicione os campos a Pivot.Fields como no Início rápido VCL.

A grade FMX desenha suas próprias barras de rolagem, exibidas enquanto são arrastadas, e rola com a roda do mouse e com o gesto de arrastar por toque. Seu método de teste de acerto é HitAt, porque o FireMonkey já publica uma propriedade HitTest.

No 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;
```

Veja também

- TsgcFMXPivotGrid
- TsgcFMXPivotFieldList
- Plataformas

Início rápido C++Builder

Instale os pacotes do sgcPivotGrid para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto e use os componentes da página de paleta **SGC PivotGrid** exatamente como no Delphi. O instalador adiciona os dois caminhos ao IDE para você. Os cabeçalhos .hpp e os arquivos .lib gerados do sgcPivotGrid e do sgcUICore estão nessa mesma pasta, portanto o sgcUICore não precisa de um caminho próprio.

O que saber em C++

- AppendRow recebe um array aberto de Variant, que o C++ passa como um array de Variant mais seu índice mais alto: `AppendRow(Row, 3)` para quatro valores.
- O cabeçalho gerado não tem argumentos padrão para os construtores de TsgcPivotEngine e TsgcPivotGridRenderer. Passe NULL explicitamente: `new TsgcPivotEngine(NULL)` e `new TsgcPivotGridRenderer(NULL, NULL)`. Veja Início rápido headless.
- Os eventos são closures, portanto seus handlers são funções membro __fastcall de uma classe. Um parâmetro var como aHandled é uma referência bool.

No código

Veja também

- TsgcVCLPivotGrid
- Início rápido VCL
- Demos

Início rápido DataSet

TsgcPivotSource_DataSet copia os registros de qualquer TDataSet para o engine de uma grade e mantém a tabela dinâmica atualizada. O layout se refere aos campos do dataset por seu FieldName, portanto pode ser definido antes de os dados chegarem.

```
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;
```

Como a origem carrega

- Columns lista os campos do dataset a carregar. Vazio carrega todos os campos. Campos BLOB, binários e aninhados são sempre ignorados.
- Os tipos de coluna vêm dos tipos de campo: campos string são texto, campos inteiros e float são números, campos de data e hora são datas, campos booleanos são booleanos.
- A carga roda com DisableControls e restaura o bookmark, de modo que os controles vinculados ao dataset não piscam.
- Com LiveUpdates, um post, um append, um delete, um refresh, First ou Last recarrega a tabela dinâmica, pois TDataSet os informa como alterações de dados. A rolagem com Next, Prior ou MoveBy não recarrega. Uma carga só roda enquanto o dataset está no estado de navegação.
- Load recarrega a qualquer momento, Refresh recarrega quando a origem está Active.

Um projeto C++Builder que usa TClientDataSet vincula midas.lib com `#pragma link "midas.lib"`, como faz a demo DataSetPivot.

Veja também

- TsgcPivotSource_DataSet
- TsgcPivotSource_Table

- Agrupamento de datas

Início rápido headless

Uma tabela dinâmica não precisa de uma janela. TsgcPivotEngine carrega as linhas e calcula, seus métodos de exportação gravam CSV, HTML e JSON, e TsgcPivotGridRenderer desenha a mesma grade dos controles em um PNG ou em um SVG. Essas units usam apenas a RTL e o sgcUICore, portanto um relatório roda em um serviço, em um servidor web ou no 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;
```

Observações

- sgcPivotRenderState retorna um estado de visão vazio: sem rolagem, sem seleção e sem substituições de largura de coluna. As sobrecargas do renderer que recebem o engine o bloqueiam enquanto desenham.
- Uma largura e uma altura 0 desenham o conteúdo inteiro. Os tamanhos estão em DIPs e o argumento de escala os multiplica.

Veja também

- Exportação
- TsgcPivotGridRenderer
- Demos

Demos

As demos são instaladas em `Demos\PivotGrid`, com os projetos Delphi em `Delphi` e suas versões para C++Builder em `CBuilder`. Toda demo aceita `--test`, que executa seu autoteste e sai, e `/snapshot <file.png>`, que salva uma imagem da tabela dinâmica.

Demo	Fra-mework	O que mostra
SalesPivot	VCL	100.000 linhas de vendas: agrupamento de datas por ano, trimestre e mês, expandir e recolher, ordenação por valor, uma escala de cores, a lista de campos e exportação para CSV, HTML e PNG.
DataSetPivot	VCL	Uma tabela dinâmica sobre um TClientDataSet: uma edição, inclusão ou exclusão no TDBGrid a recarrega, e um clique duplo em uma célula faz drill-down até os pedidos de origem.
MillionRows	VCL	1.000.000 de linhas acrescentadas em um único bloco de carga, CountDistinct de uma coluna com 100.000 valores e StdDev ao lado de Sum, com o recálculo cronometrado.
AdvancedPivot	VCL	Um campo calculado Profit, mostrar valores como, Top N com Others e o layout salvo e carregado de um arquivo JSON.
GridToPivotToChart	VCL	Pedidos digitados em um sgcGrid alimentam a tabela dinâmica por meio de TsgcPivotSource_Table, e o resultado da tabela dinâmica alimenta o sgcCharts por meio de IsgcUITable.
SalesPivotFMX	FireMonkey	50.000 linhas com o layout SalesPivot, gesto de arrastar por toque, o tema escuro e exportação PNG.
HeadlessReport	Console	Lê um arquivo CSV e grava report.png, report.svg, report.html, report.csv e report.json sem nenhum framework de interface.
AskYourPivot	VCL	Uma pergunta em linguagem natural vira o layout da tabela dinâmica por meio do TsgcPivotGridAIQuery e de um TsgcAIChat: as caixas de provedor, modelo e chave de API, Ask, Undo, perguntas de exemplo e Show prompt. Edição Registered, exige sgcWebSockets. Veja Pergunte à sua tabela dinâmica.

Veja também

- Início rápido VCL
- Início rápido headless
- Pergunte à sua tabela dinâmica

TsgcVCLPivotGrid

Unidade: sgcPivotGrid_VCL
TsgcVCLPivotGrid é a grade dinâmica VCL, somente Windows. Ela possui um TsgcPivotEngine, publicado como Pivot, e desenha seu resultado com linhas e colunas virtuais: um canto congelado, cabeçalhos de coluna de vários níveis com spans, cabeçalhos de linha, botões + e -, totais e totais gerais. Rola com barras de rolagem nativas e com a roda do mouse (Shift rola na horizontal), redimensiona uma coluna pela borda do seu cabeçalho e tem uma interface de teclado: as setas, PgUp, PgDn, Home e End movem a seleção, Right e Left (ou + e -) expandem e recolhem um membro quando o cabeçalho de linha tem o foco, e Enter faz drill-down. Herda de TsgcUIVCLControl e publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop e Visible.

Propriedades

Propriedade	Tipo	Descrição
Pivot	TsgcPivotEngine	O engine que a grade possui: seus campos, dados, totais e títulos, gravados com o formulário. A grade implementa IsgcUITable delegando a ele.
BorderStyle	TBorderStyle	bsSingle (padrão) ou bsNone.
RowLayout	TsgcPivotRowLayout	pvlCompact (padrão) desenha os níveis de linha em uma única coluna de cabeçalho com recuo, pvlTabular dá a cada nível sua própria coluna. Veja Áreas.
RowHeight	Integer	A altura de uma linha em DIPs. Padrão 24.
HeaderHeight	Integer	A altura de uma linha de cabeçalho de coluna em DIPs. Padrão 26.
IndentWidth	Integer	O recuo de cada nível de linha no layout compacto, em DIPs. Padrão 16.
MinColumnWidth, MaxColumnWidth	Integer	Os limites das larguras automáticas das colunas de dados, em DIPs. Padrões 48 e 320.
AutoColumnWidths	Boolean	Ajusta as colunas de dados ao seu conteúdo. Quando False, elas têm 96 DIPs de largura. As colunas de cabeçalho de linha sempre se ajustam. Padrão True.
GridLines	Boolean	Desenha as linhas das células. Padrão True.
AlternateRows	Boolean	Sombreia uma linha sim, outra não. Padrão False.
TotalsBold	Boolean	Desenha os totais em negrito. Padrão True.
ShowExpandButtons	Boolean	Mostra os botões `+` e `-` dos membros que têm filhos. Padrão True.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	Uma cor de destaque $AARRGGBB. 0 (padrão) usa o destaque do tema.
FontSize	Double	O tamanho da fonte das células em pontos. Padrão 12.

FormatRules	TsgcPivot-FormatRules	As regras de formatação condicional das células.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
SelectedRow	Integer	Público. O item de linha selecionado, -1 para nenhum.
SelectedCol	Integer	Público. A coluna de dados selecionada, -1 quando o cabeçalho de linha tem o foco ou nada está selecionado.
TopRow	Integer	Público. O primeiro item de linha mostrado sob os cabeçalhos de coluna.
ScrollX	Integer	Público. O deslocamento horizontal de rolagem das colunas de dados.
Renderer	TsgcPivotGridRenderer	Público, somente leitura. O renderer que desenha a grade.

Métodos

Método	Retorna	Descrição
HitTest(X, Y)	TsgcPivotHitInfo	A parte sob um ponto em pixels do dispositivo: Part (pvhNone, pvhCorner, pvhColumnHeader, pvhRowHeader, pvhCell, pvhExpandButton, pvhColumnResize), RowItem, DataCol e Level.
ScrollToCell(aRowItem, aDataCol)		Rola até que a célula fique visível. -1 deixa esse eixo como está.
Recalculate		Recalcula o engine agora.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0)		A visão inteira como imagem, tamanhos em DIPs, 0 é o tamanho do conteúdo, até 16384 pixels.
SaveToCSV, SaveToJSON, SaveToHTML(aFileName or aStream)		A visão atual como tabela. Veja Exportação.
SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile, LoadLayoutFromFile		O layout de Pivot como JSON, com as larguras de coluna da grade. Edição Advanced. Veja Persistência do layout.

Eventos

Evento	Tipo	Descrição
OnCellClick, OnCellDblClick	TsgcVCLPivotHitEvent	Um clique ou um clique duplo, com as informações de acerto do ponto.
OnDrillDown	TsgcVCLPivotDrillDownEvent	Enter ou clique duplo em uma célula: aRowItem, aDataCol e var aHandled. A grade não tem ação interna, chame Pivot.DrillDown ou Pivot.GetSourceRows para ler as linhas de origem.

OnExpand-Change	TsgcVCLPivo-tExpandEvent	Depois que um membro foi expandido ou recolhido pelo usuário: aAxis e aItem.
OnGetCellSty-le	TsgcPivot-CellStyleEvent	Antes de uma célula de dados ser desenhada: aRowItem, aDataCol e var aBack, aFore e aBold, após as regras de formatação.
OnGetCellText	TsgcPivotCell-TextEvent	Antes de uma célula de dados ser desenhada: aRowItem, aDataCol e var aText, o valor formatado, ou Pivot.EmptyText quando a célula está vazia.
OnSelection-Change	TNotifyEvent	SelectedRow ou SelectedCol mudou.
OnColumnRe-sized	TsgcVCLPi-votHitEvent	O usuário redimensionou uma coluna arrastando a borda do seu cabe-çalho.

Enquanto o formulário é carregado, o engine apenas se marca como sujo, e a grade recalcula uma vez em Loaded quando Pivot.AutoRecalculate é True.

Exemplo

```
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;
```

Veja também

- TsgcFMXPivotGrid
- TsgcVCLPivotFieldList
- Início rápido VCL

TsgcFMXPivotGrid

Unidade: sgcPivotGrid_FMX
TsgcFMXPivotGrid é a grade dinâmica FireMonkey, a partir do Delphi XE6 (SGC_PIVOTGRID_FMX). Tem o mesmo engine, layout, propriedades e eventos de TsgcVCLPivotGrid. As diferenças vêm do FireMonkey: ela desenha suas próprias barras de rolagem, exibidas enquanto são arrastadas, rola com a roda do mouse e com o gesto de arrastar por toque, e seu método de teste de acerto é HitAt, porque o FireMonkey já publica uma propriedade HitTest. Seus tipos de evento têm o prefixo TsgcFMXPivot. Herda de TsgcUIFMXControl e publica as propriedades e eventos de layout usuais do FireMonkey.

Propriedades

Propriedade	Tipo	Descrição
Pivot	TsgcPivotEngine	O engine que a grade possui: seus campos, dados, totais e títulos, gravados com o formulário. A grade implementa IsgcUITable delegando a ele.
RowLayout	TsgcPivotRowLayout	pvlCompact (padrão) desenha os níveis de linha em uma única coluna de cabeçalho com recuo, pvlTabular dá a cada nível sua própria coluna. Veja Áreas.
RowHeight	Integer	A altura de uma linha em DIPs. Padrão 24.
HeaderHeight	Integer	A altura de uma linha de cabeçalho de coluna em DIPs. Padrão 26.
IndentWidth	Integer	O recuo de cada nível de linha no layout compacto, em DIPs. Padrão 16.
MinColumnWidth, MaxColumnWidth	Integer	Os limites das larguras automáticas das colunas de dados, em DIPs. Padrões 48 e 320.
AutoColumnWidths	Boolean	Ajusta as colunas de dados ao seu conteúdo. Quando False, elas têm 96 DIPs de largura. As colunas de cabeçalho de linha sempre se ajustam. Padrão True.
GridLines	Boolean	Desenha as linhas das células. Padrão True.
AlternateRows	Boolean	Sombreia uma linha sim, outra não. Padrão False.
TotalsBold	Boolean	Desenha os totais em negrito. Padrão True.
ShowExpandButtons	Boolean	Mostra os botões + e - dos membros que têm filhos. Padrão True.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	Uma cor de destaque $AARRGGBB. 0 (padrão) usa o destaque do tema.
FontSize	Double	O tamanho da fonte das células em pontos. Padrão 12.
FormatRules	TsgcPivotFormatRules	As regras de formatação condicional das células.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.

SelectedRow	Integer	Público. O item de linha selecionado, -1 para nenhum.
SelectedCol	Integer	Público. A coluna de dados selecionada, -1 quando o cabeçalho de linha tem o foco ou nada está selecionado.
TopRow	Integer	Público. O primeiro item de linha mostrado sob os cabeçalhos de coluna.
ScrollX	Integer	Público. O deslocamento horizontal de rolagem das colunas de dados.
Renderer	TsgcPivotGridRenderer	Público, somente leitura. O renderer que desenha a grade.
VertBarVisible, HorzBarVisible	Boolean	Público, somente leitura. True quando o conteúdo precisa dessa barra de rolagem.

Métodos

Método	Retorna	Descrição
HitAt(X, Y)	TsgcPivotHitInfo	A parte sob um ponto em coordenadas locais: Part (pvhNone, pvhCorner, pvhColumnHeader, pvhRowHeader, pvhCell, pvhExpandButton, pvhColumnResize), RowItem, DataCol e Level.
ScrollToCell(aRowItem, aDataCol)		Rola até que a célula fique visível. -1 deixa esse eixo como está.
Recalculate		Recalcula o engine agora.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0)		A visão inteira como imagem, tamanhos em DIPs, 0 é o tamanho do conteúdo, até 16384 pixels.
SaveToCSV, SaveToJSON, SaveToHTML(aFileName or aStream)		A visão atual como tabela. Veja Exportação.
SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile, LoadLayoutFromFile		O layout de Pivot como JSON, com as larguras de coluna da grade. Edição Advanced. Veja Persistência do layout.

Eventos

Evento	Tipo	Descrição
OnCellClick, OnCellDbl-Click	TsgcFMXPi-votHitEvent	Um clique ou um clique duplo, com as informações de acerto do ponto.
OnDrillDown	TsgcFMXPivot-DrillDownEvent	Enter ou clique duplo em uma célula: aRowItem, aDataCol e var aHan-dled. A grade não tem ação interna, chame Pivot.DrillDown ou Pivot.GetSourceRows para ler as linhas de origem.
OnExpand-Change	TsgcFMXPivo-tExpandEvent	Depois que um membro foi expandido ou recolhido pelo usuário: aAxis e aItem.
OnGetCellSty-le	TsgcPivot-CellStyleEvent	Antes de uma célula de dados ser desenhada: aRowItem, aDataCol e var aBack, aFore e aBold, após as regras de formatação.
OnGetCellText	TsgcPivotCell-TextEvent	Antes de uma célula de dados ser desenhada: aRowItem, aDataCol e var aText, o valor formatado, ou Pivot.EmptyText quando a célula está vazia.
OnSelection-Change	TNotifyEvent	SelectedRow ou SelectedCol mudou.
OnColumnRe-sized	TsgcFMXPi-votHitEvent	O usuário redimensionou uma coluna arrastando a borda do seu cabe-çalho.

Veja também

- TsgcVCLPivotGrid
- TsgcFMXPivotFieldList
- Início rápido FMX

TsgcVCLPivotFieldList

Unidade: sgcPivotGrid_FieldList_VCL
TsgcVCLPivotFieldList é a lista de campos VCL. Vinculada a uma grade dinâmica, permite que o usuário monte o layout arrastando e soltando. Herda de TsgcUIVCLControl e publica as propriedades usuais de layout da VCL. Sua lista de verificação de membros é um TsgcVCLPivotMemberPopup e seu menu de agregação é um TPopupMenu.

Como o usuário trabalha com ela

- As colunas de origem são listadas no topo, com uma marca para as colunas que estão no layout. Abaixo há quatro zonas: Filtros, Colunas, Linhas e Valores.
- Arraste uma coluna para uma zona para adicionar um campo. Arraste um chip para outra zona para movê-lo, ou para fora das zonas para removê-lo.
- Uma coluna solta em Valores recebe Count quando é uma coluna de texto e Sum nos demais casos. O botão de um chip de Valores abre um menu com todas as agregações.
- O botão de filtro de um chip de Filtros, Colunas ou Linhas abre a lista de verificação de membros, com uma caixa de pesquisa e Selecionar tudo. OK grava FilterValues e FilterExclude, veja Filtros.
- Os filtros de rótulo e os filtros de valor são definidos no código ou no Object Inspector.

Propriedades

Propriedade	Tipo	Descrição
PivotGrid	TComponent	A grade dinâmica cujo engine a lista edita, um TsgcVCLPivotGrid ou um TsgcFMXPivotGrid.
Engine	TsgcPivotEngine	Público. Um engine definido no código tem prioridade sobre o engine de PivotGrid, para um engine headless.
ShowFilterArea	Boolean	Mostra a zona Filtros. Padrão True.
DeferUpdate	Boolean	Reúne as alterações do usuário e as aplica somente em ApplyChanges, com um único recálculo. Padrão False.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	Uma cor de destaque $AARRGGBB. 0 (padrão) usa o destaque do tema.
FontSize	Double	O tamanho da fonte em pontos.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.

Métodos

Método	Retorna	Descrição
ApplyChanges		Atribui o layout pendente ao engine quando DeferUpdate é True.

Eventos

Evento	Tipo	Descrição
OnBeforeFieldMove	TsgcPivotBeforeFieldMoveEvent	Antes de o usuário mover um campo: aFieldName, aFrom, aTo e var aAllow. Para uma coluna de origem solta em uma zona, aFrom é igual a aTo. Remover um campo não gera evento de movimento.
OnFieldMoved	TsgcPivotFieldMovedEvent	Depois que um campo foi adicionado ou movido: aField.

Exemplo

```
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;
```

Veja também

- TsgcFMXPivotFieldList
- Áreas

TsgcFMXPivotFieldList

Unidade: sgcPivotGrid_FieldList_FMX
TsgcFMXPivotFieldList é a lista de campos FireMonkey, a partir do Delphi XE6 (SGC_PIVOTGRID_FMX), com as mesmas zonas, o mesmo arrastar e soltar e as mesmas regras de filtro de TsgcVCLPivotFieldList. Seus popups abrem quando o botão é solto sobre o mesmo chip.

Como o usuário trabalha com ela

- As colunas de origem são listadas no topo, com uma marca para as colunas que estão no layout. Abaixo há quatro zonas: Filtros, Colunas, Linhas e Valores.
- Arraste uma coluna para uma zona para adicionar um campo. Arraste um chip para outra zona para movê-lo, ou para fora das zonas para removê-lo.
- Uma coluna solta em Valores recebe Count quando é uma coluna de texto e Sum nos demais casos. O botão de um chip de Valores abre um menu com todas as agregações.
- O botão de filtro de um chip de Filtros, Colunas ou Linhas abre a lista de verificação de membros, com uma caixa de pesquisa e Selecionar tudo. OK grava FilterValues e FilterExclude, veja Filtros.
- Os filtros de rótulo e os filtros de valor são definidos no código ou no Object Inspector.

Propriedades

Propriedade	Tipo	Descrição
PivotGrid	TComponent	A grade dinâmica cujo engine a lista edita, um TsgcVCLPivotGrid ou um TsgcFMXPivotGrid.
Engine	TsgcPivotEngine	Público. Um engine definido no código tem prioridade sobre o engine de PivotGrid, para um engine headless.
ShowFilterArea	Boolean	Mostra a zona Filtros. Padrão True.
DeferUpdate	Boolean	Reúne as alterações do usuário e as aplica somente em ApplyChanges, com um único recálculo. Padrão False.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom.
AccentColor	TsgcUIColor	Uma cor de destaque $AARRGGBB. 0 (padrão) usa o destaque do tema.
FontSize	Double	O tamanho da fonte em pontos.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
Model	TsgcPivotFieldListModel	Público, somente leitura. O modelo independente de framework com as zonas, os chips e o estado do arrastar.
AggregationMenu, FilterPopup, FilterSearch, FilterSelectAll, FilterList	controles de menu e popup	Público, somente leitura. O menu de agregação e as partes da lista de verificação de membros, criados na primeira vez em que são exibidos.

Métodos

Método	Retorna	Descrição
ApplyChanges		Atribui o layout pendente ao engine quando DeferUpdate é True, com um único recálculo.
HitAt(X, Y)	TsgcPivotFieldListHitInfo	A parte da lista sob um ponto local.
ShowAggregationMenu(aField)		Abre o menu de agregação de um campo de Valores.
ShowMemberFilter(aField)		Abre a lista de verificação de membros de um campo de Filtros, Colunas ou Linhas.
CloseMemberFilter(aApply)		Fecha a lista de verificação de membros. aApply grava os membros marcados no filtro.

Eventos

Evento	Tipo	Descrição
OnBeforeFieldMove	TsgcPivotBeforeFieldMoveEvent	Antes de o usuário mover um campo: aFieldName, aFrom, aTo e var aAllow. Para uma coluna de origem solta em uma zona, aFrom é igual a aTo. Remover um campo não gera evento de movimento.
OnFieldMoved	TsgcPivotFieldMovedEvent	Depois que um campo foi adicionado ou movido: aField.

Veja também

- TsgcVCLPivotFieldList
- TsgcFMXPivotGrid

TsgcPivotEngine

Unidade: sgcPivotGrid_Engine
TsgcPivotEngine mantém as linhas de origem em um armazenamento de colunas tipado, o layout e o resultado calculado. As grades possuem um, publicado como Pivot, e ele também funciona sozinho, sem nenhuma interface. Descende de TPersistent, implementa IsgcUITable sem contagem de referências e é criado com `Create(aOwner: TPersistent = nil)`. Em C++ passe NULL explicitamente.

Propriedades

Propriedade	Tipo	Descrição
Campos	TsgcPivotFields	Os campos do layout, nas quatro áreas.
CalculatedFields	TsgcPivotCalculatedFields	Os campos calculados. Edição Advanced.
RowGrandTotals, ColumnGrandTotals	Boolean	Mostra a linha de total geral e a coluna de total geral. Padrão True.
SubtotalPosition	TsgcPivotSubtotalPosition	pvpTop (padrão) ou pvpBottom. Veja Totais.
AutoRecalculate	Boolean	Recalcula após cada alteração dos dados ou do layout, uma vez por bloco. Quando False, chame Recalculate. Padrão True.
GrandTotalCaption	string	O título dos totais gerais. Padrão Grand Total.
BlankCaption	string	O rótulo do membro vazio. Padrão (blank).
EmptyText	string	O texto de uma célula de dados sem valor, vazio por padrão. As grades, o renderer, a exportação HTML e ResultTable o exibem, e OnGetCellText o recebe e pode substituí-lo. SaveToCSV grava um campo vazio e SaveToJSON grava null. Uma alteração republica a visão sem recalcular.
OthersCaption	string	O rótulo do membro Others do Top N. Padrão Others. Edição Advanced.
MaxCells	Integer	O maior número de células que um layout pode criar. Um recálculo que precise de mais gera EsgcPivotError. Padrão 5.000.000.
Result	TsgcPivotResult	Público, somente leitura. O snapshot atual, nil antes do primeiro cálculo. Leia-o somente entre Lock e Unlock.
ResultTable	TsgcPivotResultTable	Público, somente leitura. A visão IsgcUITable do resultado. Veja Exportação.
Store	TsgcPivotStore	Público, somente leitura. O armazenamento de colunas, usado pelas origens.
RowLayout	TsgcPivotRowLayout	Público. Definido pelas grades: no layout compacto, um item pai vem antes de seus filhos.

Métodos: dados

Método	Retorna	Descrição
BeginLoad(aClear = True), EndLoad		Um bloco de carga. aClear esvazia primeiro o armazenamento. O engine recalcula uma vez, no EndLoad mais externo.
DefineColumn(aName), DefineColumn(aName, aType)	Integer	Adiciona uma coluna. Sem tipo, o primeiro valor que não seja vazio o define. Os nomes são únicos, ignorando maiúsculas e minúsculas.
AppendRow(aValues)	Integer	Acrescenta uma linha de valores Variant na ordem das colunas. Null e valores que não podem ser convertidos ficam vazios.
ClearData		Remove todas as linhas e colunas.
LoadFromTable(aTable)		Carrega todas as colunas e linhas de uma IsgcUITable.
SourceRowCount, SourceColumnCount	Integer	O tamanho do armazenamento.
SourceColumnName(aIndex)	string	O nome de uma coluna.
SourceValue(aRow, aCol)	Variant	Um valor armazenado, Null quando vazio.
GetMemberLabels(aField, aList)		Os rótulos de membro com os quais um filtro de membros de aField compara, na ordem de exibição, com BlankCaption por último quando um valor está vazio.

Métodos: layout e visão

Método	Retorna	Descrição
BeginUpdate, EndUpdate		Um bloco de layout com um único recálculo no EndUpdate.
Recalculate		Recalcula agora.
AddField(aFieldName, aArea, aGrouping = pvdNone)	TsgcPivotField	Adiciona um campo de linha, coluna, filtro ou dados ao final de sua área.
AddDataField(aFieldName, aAggregation = pvfSum)	TsgcPivotField	Adiciona uma medida.
MoveField(aField, aArea, aIndex)		Move um campo para outra área ou posição.
RemoveField(aField), ClearLayout		Remove um campo, ou todos os campos.
ExpandAll(aAxis), CollapseAll(aAxis)		Expande ou recolhe todos os membros de pvxRows ou pvxColumns.
ExpandLevel(aAxis, aLevel), CollapseLevel(aAxis, aLevel)		Expande ou recolhe todos os membros de um nível.

SetExpanded(aAxis, aItem, aValue), ToggleExpanded(aAxis, aItem)		Altera um item do snapshot atual. O estado de expansão é mantido por caminho do membro, portanto sobrevive a um recálculo. Veja Expandir e recolher.
GetExpandState(aAxis, aState), SetExpandState(aAxis, aState)		O estado de expansão explícito de um eixo como uma lista de strings. Edição Advanced.

Métodos: resultado, drill-down e exportação

Método	Retorna	Descrição
Lock, Unlock		Protegem o snapshot enquanto Result é lido.
GetSourceRows(aRowItem, aDataCol, var aRows)	Integer	As linhas de origem por trás de uma célula, em ordem de linha. aDataCol -1 usa todas as colunas, e um total usa todas as linhas abaixo dele.
DrillDown(aRowItem, aDataCol)	TsgcPivotSourceView	As mesmas linhas como uma IsgcUITable com todas as colunas do armazenamento. Quem chama a libera, antes do engine. Ela fica sem linhas depois que os dados mudam.
SaveToCSV, SaveToJSON, SaveToHTML		A visão atual, arquivo ou stream. Veja Exportação.
SaveLayoutToString, LoadLayoutFromString, SaveLayoutToFile, LoadLayoutFromFile		O layout como JSON. Edição Advanced. Veja Persistência do layout.
Assign(Source)		Copia o layout e as configurações de outro engine, não seus dados.

Eventos

Evento	Tipo	Descrição
OnCalculated	TNotifyEvent	Depois que um novo resultado foi publicado, na thread que recalculou.
OnLayoutChange	TNotifyEvent	Depois que os campos ou as configurações do layout mudaram.
OnFilterRow	TsgcPivotFilterRowEvent	Para cada linha de origem enquanto atribuído: aRow e var aAccept. Veja Filtros.

Exemplo

Leitura do resultado: os rótulos das linhas e os textos das células, entre Lock e 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;
```

Veja também

- TsgcPivotField
- Início rápido headless
- Arquitetura

TsgcPivotField

Unidade: sgcPivotGrid_Fields
TsgcPivotField é um item de TsgcPivotEngine.Fields (TsgcPivotFields), um campo do layout. O mesmo FieldName pode aparecer em vários campos.

Propriedades

Propriedade	Tipo	Descrição
FieldName	string	A coluna de origem ou o campo calculado.
Caption	string	O título do nível ou da medida. Vazio usa DisplayCaption.
DisplayCaption	string	Público, somente leitura. Caption, senão Sum of Sales para um campo de dados, senão FieldName.
Area	TsgcPivotArea	pvaRow (padrão), pvaColumn, pvaData ou pvaFilter. Veja Áreas.
AreaIndex	Integer	A posição na área. Para um campo de dados, é o índice da medida.
DateGrouping	TsgcPivotDateGrouping	pvdNone (padrão), pvdYear, pvdQuarter, pvdMonth, pvdWeek ou pvdDay. Veja Agrupamento de datas.
DisplayFormat	string	Uma máscara FormatDateTime ou FormatFloat para os rótulos dos membros, ou para os valores de um campo de dados.
Aggregation	TsgcPivotAggregation	A função de um campo de dados, pvfSum por padrão. Veja Agregações.
SortBy	TsgcPivotSortBy	pvsLabel (padrão) ordena os membros pelo rótulo, números e datas pelo valor, pvsValue pelo seu total.
SortOrder	TsgcPivotSortOrder	pvoAscending (padrão) ou pvoDescending.
SortMeasure	Integer	O campo de dados, por AreaIndex, de uma ordenação por valor e do Top N. Padrão 0.
Subtotals	Boolean	Mostra os subtotais dos membros. Padrão True.
Expanded	Boolean	O estado de expansão inicial dos membros deste nível. Padrão True.
FilterValues	TStrings	O filtro de membros. Veja Filtros.
FilterExclude	Boolean	FilterValues lista os membros a excluir. Padrão False.
LabelFilter	TsgcPivotLabelFilter	Operator, Value e Value2 sobre os rótulos dos membros.
ValueFilter	TsgcPivotValueFilter	Operator, Value, Value2 e Measure sobre os totais dos membros.
TopN, TopNOthers	Integer, Boolean	Top N com um membro Others opcional. Edição Advanced. Veja Top N.

ShowAs, ShowAsAxis, ShowAsBaseItem	TsgcPivotShowAs, TsgcPivotAxisKind, string	Mostrar valores como. Edição Advanced. Veja Mostrar valores como.

Métodos

Método	Retorna	Descrição
MoveTo(aArea, aAreaIndex)		Move o campo com uma única notificação de alteração. MaxInt é o fim da área.
Assign(Source)		Copia todas as propriedades, inclusive Area e AreaIndex.

TsgcPivotFields

Método	Retorna	Descrição
Add	TsgcPivotField	Adiciona um campo.
AreaCount(aArea)	Integer	O número de campos em uma área.
FieldAt(aArea, aAreaIndex)	TsgcPivotField	O campo em uma posição de uma área.
Items[aIndex]	TsgcPivotField	Os campos na ordem da coleção.

Dentro de BeginUpdate e EndUpdate, e enquanto um formulário é carregado, os valores de AreaIndex são mantidos como foram dados. EndUpdate renumera cada área por AreaIndex e depois pela ordem da coleção.

Veja também

- TsgcPivotEngine
- TsgcPivotCalculatedField

TsgcPivotCalculatedField

Unidade: sgcPivotGrid_Fields
TsgcPivotCalculatedField é um item de TsgcPivotEngine.CalculatedFields (TsgcPivotCalculatedFields): uma coluna Double calculada por linha de origem a partir de uma expressão, que os campos usam por meio de FieldName = Name como uma coluna de origem. Edição Advanced, veja Campos calculados para a sintaxe.

Propriedades

Propriedade	Tipo	Descrição
Name	string	O nome ao qual os campos se referem. Não pode ser o nome de uma coluna de origem.
Expression	string	A expressão, por exemplo [Revenue] - [Cost].

TsgcPivotCalculatedFields

Método	Retorna	Descrição
Add	TsgcPivotCalculatedField	Adiciona um campo calculado.
IndexOfName(aName)	Integer	O índice de um campo calculado, -1 quando não há nenhum.
Items[aIndex]	TsgcPivotCalculatedField	Os campos calculados.

Veja também

- Campos calculados
- TsgcPivotField

TsgcPivotSource_DataSet

Unidade: sgcPivotGrid_Source_DataSet
TsgcPivotSource_DataSet é um componente não visual que copia um TDataSet para um engine dinâmico por meio de um TDataLink e o recarrega quando os dados mudam. Veja Início rápido DataSet.

Propriedades

Propriedade	Tipo	Descrição
DataSet	TDataSet	O dataset a carregar.
Active	Boolean	Carrega os dados agora e os mantém atualizados. Padrão False.
Columns	TStrings	As colunas a carregar. Vazio carrega todas as colunas.
LiveUpdates	Boolean	Recarrega quando os dados mudam enquanto Active. Padrão True.
PivotGrid	TCompo-nent	A grade dinâmica cujo engine recebe os dados.
Engine	TsgcPivo-tEngine	Público. O engine no qual a origem grava: um engine atribuído no código, senão o engine de PivotGrid.

Métodos

Método	Retorna	Descrição
Load		Carrega o dataset agora, Active ou não.
Refresh		Recarrega quando Active.
FieldTypeToDataType(aFieldType)	TsgcPivotDa-taType	Função de classe. O tipo de coluna de um tipo de campo DB.

- A carga usa DisableControls e restaura o bookmark. Campos BLOB, binários e aninhados são ignorados.
- Com LiveUpdates, a tabela dinâmica recarrega quando o dataset informa uma alteração de dados no estado de navegação: um post, um append, um delete, um refresh, First ou Last. Next, Prior e MoveBy não recarregam.

Veja também

- TsgcPivotSource_Table
- Início rápido DataSet

TsgcPivotSource_Table

Unidade: sgcPivotGrid_Source_Table
TsgcPivotSource_Table copia qualquer IsgcUITable para um engine dinâmico: um TsgcGridData, um TsgcVCL-Grid ou TsgcFMXGrid do sgcGrid, o resultado de outra tabela dinâmica ou sua própria classe. Ele escuta a tabela e a recarrega, sob TableLock, quando TableVersion mudou. A recarga roda na thread que notifica a alteração.

Propriedades

Propriedade	Tipo	Descrição
Table	TComponent	Um componente que implementa IsgcUITable.
Active	Boolean	Carrega os dados agora e os mantém atualizados. Padrão False.
Columns	TStrings	As colunas a carregar. Vazio carrega todas as colunas.
LiveUpdates	Boolean	Recarrega quando os dados mudam enquanto Active. Padrão True.
PivotGrid	TComponent	A grade dinâmica cujo engine recebe os dados.
Engine	TsgcPivotEngine	Público. O engine no qual a origem grava: um engine atribuído no código, senão o engine de PivotGrid.

Métodos

Método	Retorna	Descrição
SetTable(aTable)		Vincula qualquer objeto que implemente IsgcUITable. nil desvincula.
GetTable	IsgcUITable	A tabela vinculada.
Load		Carrega a tabela agora, Active ou não, mesmo que sua versão não tenha mudado.
Refresh		Recarrega quando Active.

Colunas com o mesmo nome de uma coluna anterior são ignoradas.

Exemplo

```
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;
```

Veja também

- TsgcPivotSource_DataSet
- Exportação

TsgcPivotGridRenderer

Unidade: sgcPivotGrid_Render
TsgcPivotGridRenderer desenha um resultado da tabela dinâmica em qualquer canvas do sgcUICore, e é o que as grades VCL e FMX usam. Sozinho, produz a saída headless em PNG e SVG. Crie-o com `Create(aOptions: TsgcPivotRenderOptions = nil; aFormatRulesOwner: TPersistent = nil)`. Em C++ passe NULL para ambos. Veja Início rápido headless.

Propriedades

Propriedade	Tipo	Descrição
Opções	TsgcPivotRenderOptions	RowLayout, RowHeight, HeaderHeight, IndentWidth, MinColumnWidth, MaxColumnWidth, AutoColumnWidths, GridLines, AlternateRows, TotalsBold, ShowExpandButtons, Theme, AccentColor e FontSize, com o mesmo significado e os mesmos padrões de TsgcVCLPivotGrid.
FormatRules	TsgcPivotFormatRules	As regras de formatação condicional.
GrandTotalCaption, TotalCaption	string	Os títulos dos totais gerais e dos subtotais inferiores. As sobrecargas de SaveToPNG e SaveToSVG que recebem o engine copiam GrandTotalCaption do engine.
CurrentResult	TsgcPivotResult	Somente leitura. O resultado do último Render.

Métodos

Método	Retorna	Descrição
Render(aResult, aState, aCanvas, aWidth, aHeight, aScale)		Desenha a parte visível de um resultado, aWidth e aHeight em pixels do dispositivo, aScale pixels do dispositivo por DIP.
SaveToPNG, SaveToSVG(aResult or aEngine, aState, aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 1)		O conteúdo inteiro como imagem, 0 é o tamanho do conteúdo. As sobrecargas com engine bloqueiam o engine enquanto desenham.
HitTest(aResult, aState, aX, aY)	TsgcPivotHitInfo	A parte sob um ponto.
ContentWidth, ContentHeight	Integer	O tamanho do conteúdo inteiro em DIPs.
FrozenWidth, FrozenHeight	Integer	O tamanho dos cabeçalhos de linha e dos cabeçalhos de coluna.
VisibleRange	Boolean	Os primeiros e últimos itens de linha e colunas de dados visíveis de um viewport.

ColumnKey(aResult, aDataCol)	string	A chave de uma coluna de dados, usada pelas substituições de largura de coluna.

Eventos

Evento	Tipo	Descrição
OnGetCellStyle	TsgcPivotCellStyleEvent	As cores e o estado de negrito de uma célula de dados, após as regras de formatação.
OnGetCellText	TsgcPivotCellTextEvent	O texto de uma célula de dados: o valor formatado, ou o EmptyText do resultado quando a célula está vazia.

Auxiliares

Rotina	Descrição
sgcPivotRenderState	Um TsgcPivotRenderState vazio: sem rolagem, seleção, hover nem substituições de largura de coluna.
sgcPivotSetColumnWidth(var aState, aKey, aWidth)	Define a largura em DIPs da coluna de dados cuja ColumnKey é aKey.

Veja também

- Início rápido headless
- TsgcPivotFormatRule
- Exportação

TsgcPivotFormatRule

Unidade: sgcPivotGrid_Format
TsgcPivotFormatRule é um item de FormatRules (TsgcPivotFormatRules) das grades e do renderer: uma regra de valor que colore as células cujo valor satisfaz uma condição, ou uma escala de cores entre duas cores.

Propriedades

Propriedade	Tipo	Descrição
Kind	TsgcPivotFormatRuleKind	pvrValue (padrão), uma regra de valor, ou pvrColorScale.
Measure	Integer	O campo de dados, por AreaIndex, ao qual a regra se aplica. -1 (padrão) aplica a todas as medidas.
Operator, Value, Value2	TsgcPivotFilterOperator, Double, Double	A condição, com os operadores dos filtros. pvcNone satisfaz todo valor que não é vazio. Uma escala de cores também a usa para selecionar seus valores.
BackColor, ForeColor	TsgcUIColor	As cores $AARRGGBB de uma célula que satisfaz a regra. 0 (padrão) mantém a cor. Uma escala de cores usa apenas ForeColor.
Bold	Boolean	Desenha uma célula que satisfaz a regra em negrito. Padrão False.
MinColor, MaxColor	TsgcUIColor	As cores de uma escala no menor e no maior valor das células de membro da medida. Uma MinColor transparente se mistura gradualmente a MaxColor.
Scope	TsgcPivotFormatScopes	As células às quais a regra se aplica: pveCells (padrão, células de membro), pveTotals e pveGrandTotals.
Enabled	Boolean	Padrão True.
StopOnMatch	Boolean	Uma correspondência encerra a avaliação das regras para aquela célula. Padrão False.

Avaliação

- As regras são executadas na ordem da coleção. Toda regra habilitada que satisfaz aplica suas cores diferentes de 0, de modo que uma correspondência posterior substitui uma anterior, e Bold se acumula.
- Uma célula vazia não satisfaz nenhuma regra e nunca é colorida.
- O intervalo de uma escala de cores é tomado sobre as células de membro, de modo que os totais fora dele recebem as cores das extremidades.
- OnGetCellStyle é executado depois das regras e pode alterar o resultado.

Exemplo

```
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;
```

Veja também

- TsgcVCLPivotGrid
- TsgcPivotGridRenderer

Áreas

Todo campo do layout fica em uma de quatro áreas. A propriedade Area de TsgcPivotField e o segundo argumento de AddField a escolhem.

Area	Significado
pvaRow	Os membros se tornam os cabeçalhos de linha. Vários campos de linha se aninham, o primeiro é o nível mais externo.
pvaCo-lumn	Os membros se tornam os cabeçalhos de coluna, aninhados da mesma forma.
pvaData	Uma medida: os valores agregados em cada célula. Vários campos de dados geram várias colu-nas de medida sob cada membro de coluna.
pvaFilter	Um filtro de relatório. Seus membros não são exibidos, ele apenas restringe as linhas de origem por meio de seus filtros de membro e de rótulo.

Ordem e movimentação

- AreaIndex ordena os campos de uma área. AddField, AddDataField e a definição de Area colocam um cam-po no fim da nova área.
- MoveField(aField, aArea, aIndex) move um campo para outra área ou posição com uma única notificação. RemoveField o exclui e ClearLayout remove todos os campos.
- Fields.AreaCount(aArea) e Fields.FieldAt(aArea, aIndex) leem o layout área por área.
- O mesmo FieldName pode aparecer em vários campos, por exemplo uma coluna Date agrupada por ano e por mês, ou uma coluna Sales com Sum e Average.
- O índice de medida usado por SortMeasure, ValueFilter.Measure e pelas regras de formatação é o AreaIn-dex do campo de dados.

Nomes desconhecidos

Um campo de linha, coluna ou filtro cujo FieldName não é uma coluna de origem nem um campo calculado é igno-rado. Um campo de dados com nome desconhecido mantém uma medida vazia, de modo que os índices de medi-da não se deslocam. Isso permite definir um layout antes de os dados serem carregados.

Linhas compactas e tabulares

RowLayout das grades decide como os níveis de linha são desenhados. pvlCompact (o padrão) mostra todos os níveis em uma única coluna de cabeçalho de linha com recuo, com cada pai em sua própria linha. pvlTabular dá a cada nível de linha sua própria coluna de cabeçalho.

Exemplo

```
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;
```

Toda alteração do layout dispara OnLayoutChange do engine. Dentro de BeginUpdate e EndUpdate o engine re-calcula uma vez, no EndUpdate.

Veja também

- Agregações
- TsgcPivotEngine
- TsgcVCLPivotFieldList

Agregações

Um campo de dados agrega sua coluna de origem com a função em Aggregation. Valores vazios são ignorados por todas as funções.

Aggregation	Result
pvfSum	A soma dos valores. O padrão.
pvfCount	O número de valores que não são vazios, como no Excel. Funciona em todos os tipos de coluna.
pvfAverage	Sum dividido por Count.
pvfMin, pvfMax	O menor e o maior valor.
pvfCountDistinct	O número de valores diferentes que não são vazios. Em uma coluna de texto, conta rótulos diferentes.
pvfStdDev, pvfVar	O desvio padrão e a variância amostrais (`n - 1`). Uma célula com menos de dois valores fica vazia.

Os totais são exatos

Subtotais e totais gerais são calculados a partir dos valores de origem, não a partir das células abaixo deles. A média de uma região é a média de suas linhas, não a média das médias de seus produtos, e uma contagem distinta de um total conta cada valor uma única vez.

Títulos e formatos

- Sem Caption, um campo de dados é exibido como *<agregação> de <FieldName>*, por exemplo Sum of Sales. DisplayCaption retorna o título em uso.
- DisplayFormat é uma máscara FormatFloat para os valores da medida, por exemplo `#,##0.00`.
- As medidas ficam sempre nas colunas. Com vários campos de dados, as colunas de medida formam o nível de coluna mais interno.

A lista de campos adiciona Count quando o usuário solta uma coluna de texto em Valores e Sum nos demais casos, e seus chips de Valores abrem um menu com todas as agregações.

Exemplo

```
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;
```

Veja também

- Mostrar valores como
- Totais
- TsgcPivotField

Agrupamento de datas

Um campo de linha, coluna ou filtro sobre uma coluna de data agrupa as datas com DateGrouping, o terceiro argumento de AddField.

DateGrouping	Rótulo do membro	Rótulo sob um nível mais amplo do mesmo campo
pvdNone	Cada valor diferente de data e hora é um membro próprio.	
pvdYear	2024	
pvdQuarter	2024 Q1	Q1
pvdMonth	2024-01	Jan, o nome abreviado do mês
pvdWeek	2024-W05, a semana ISO 8601 e seu ano	W05
pvdDay	A data curta	

- Quando o nível anterior do mesmo eixo é a mesma coluna com um agrupamento mais amplo, o rótulo omite o ano, como mostra a última coluna. Year, Quarter e Month de uma mesma coluna Date dão 2024, Q1, Jan.
- Com um DisplayFormat, o rótulo é o início do grupo formatado com FormatDateTime e essa máscara.
- Os membros de data são ordenados por data, não por rótulo, de modo que 2024-10 vem depois de 2024-09.
- Uma linha com data vazia pertence ao membro em branco, exibido com BlankCaption e ordenado por último.

Exemplo

```
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 é o estado inicial de um nível: aqui todo trimestre começa recolhido e o usuário abre os meses de um trimestre com o botão +.

Veja também

- Áreas

- Totais

Totais

Uma tabela dinâmica mostra um subtotal para cada membro que tem filhos e um total geral para cada eixo.

Propriedade	Efeito
TsgcPivotEngine.RowGrandTotals	Mostra a linha de total geral. Padrão True.
TsgcPivotEngine.ColumnGrandTotals	Mostra a coluna de total geral. Padrão True.
TsgcPivotEngine.SubtotalPosition	pvpTop (padrão): o subtotal fica no próprio membro, antes de seus filhos, e em um layout compacto a linha do pai o contém. pvpBottom: uma linha ou coluna de total segue os filhos, com o rótulo *<membro> Total*.
TsgcPivotField.Subtotals	False oculta os subtotais dos membros desse campo. Padrão True.
TsgcPivotEngine.GrandTotalCaption	O título dos totais gerais, Grand Total por padrão.
TsgcPivotEngine.BlankCaption	O rótulo do membro dos valores vazios, (blank) por padrão.

- Os totais são calculados a partir das linhas de origem com mesclagens exatas, de modo que uma média ou uma contagem distinta de um total está correta. Veja Agregações.
- Após um filtro de valor ou Top N, os totais incluem apenas os membros que permanecem visíveis, como no Excel.
- As células de total geral são sempre calculadas, mesmo quando a linha ou a coluna de total geral está oculta, de modo que mostrar valores como continua tendo seus valores base.
- O renderer e as grades desenham os totais em negrito enquanto TotalsBold é True, e as regras de formatação os selecionam com os escopos pveTotals e pveGrandTotals.

Exemplo

```
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;
```

Veja também

- Filtros
- TsgcPivotEngine

Filtros

Quatro filtros reduzem o que uma tabela dinâmica mostra. Os três primeiros são propriedades de um campo, o quarto é um evento do engine.

Filtro	Aplica-se a	Como funciona
Member filter: FilterValues, FilterExclude	Campos de linha, coluna e filtro	FilterValues lista rótulos de membro, comparados com distinção entre maiúsculas e minúsculas, como são exibidos (após DateGrouping e DisplayFormat). As linhas de origem devem ter um deles, ou nenhum deles com FilterExclude. Vazio mantém todas as linhas.
LabelFilter	Campos de linha, coluna e filtro	Operator, Value e Value2 comparados com o rótulo do membro, ignorando maiúsculas e minúsculas. Quando os dois lados são números, são comparados como números.
ValueFilter	Campos de linha e de coluna	Operator, Value e Value2 comparados com o total de cada membro para o campo de dados cujo AreaIndex é Measure. É executado nível por nível, dentro de cada pai. Um total vazio passa somente com pvcNone.
OnFilterRow	Linhas de origem	Gerado para cada linha de origem enquanto atribuído. Defina aAccept como False para excluir a linha.

Os filtros de membro e de rótulo removem linhas de origem, portanto alteram todos os totais. O filtro de valor remove membros após a agregação, e os totais são calculados novamente sobre os membros que permanecem.

Operadores

Operador	Satisfaz quando
pvcNone	Sempre: nenhum filtro.
pvcEqual, pvcNotEqual	O valor é igual a Value, ou não é.
pvcGreater, pvcGreaterEqual, pvcLess, pvcLessEqual	O valor se compara com Value.
pvcBetween, pvcNotBetween	O valor está entre Value e Value2, ambos incluídos, em qualquer ordem, ou não está.
pvcContains, pvcNotContains, pvcBeginsWith, pvcEndsWith	Testes de texto do rótulo (LabelFilter) ou do texto invariante do valor.

Listas de membros

TsgcPivotEngine.GetMemberLabels(aField, aList) preenche uma lista com os rótulos com os quais FilterValues é comparado, na ordem de exibição. O popup de membros das listas de campos o usa: OK armazena a lista mais curta, os membros marcados, ou os desmarcados com FilterExclude, e marcar todos os membros limpa o filtro.

Exemplo

```
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;
```

Veja também

- Top N
- Totais
- TsgcPivotField

Top N

TopN mantém, dentro de cada pai, os N membros de um campo de linha ou de coluna com o maior total. É um recurso avançado, que não faz parte da edição Community.

Propriedade	Efeito
TsgcPivotField.TopN	O número de membros a manter. 0 (padrão) mantém todos os membros.
TsgcPivotField.SortMeasure	O campo de dados, por seu AreaIndex, cujos totais classificam os membros. É também a medida de uma ordenação por valor.
TsgcPivotField.TopNOthers	True adiciona um membro Others com o total dos membros excluídos. Ele vem sempre por último.
TsgcPivotEngine.OthersCaption	O rótulo do membro Others, Others por padrão.

- O Top N decide quais membros permanecem. SortBy e SortOrder decidem sua ordem, de modo que pvsValue com pvoDescending os lista do maior para o menor.
- Os totais dos pais e os totais gerais incluem apenas os membros mantidos, mais Others quando ele é exibido.
- Um drill-down em Others retorna as linhas de origem de todos os membros que ele contém.
- Um filtro de valor nunca cria um membro Others.

Exemplo

```
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;
```

Veja também

- Filtros
- Edições

Mostrar valores como

ShowAs altera como os valores de um campo de dados são exibidos, sem alterar os próprios valores: RawValue, os totais, a ordenação, os filtros e o drill-down continuam usando os números reais. É um recurso avançado, que não faz parte da edição Community.

ShowAs	Valor exibido
pvwValue	O próprio valor. O padrão.
pvwPercentOfRow	O valor dividido pelo total de sua linha, como porcentagem.
pvwPercentOfColumn	O valor dividido pelo total de sua coluna, como porcentagem.
pvwPercentOfGrandTotal	O valor dividido pelo total geral, como porcentagem.
pvwRunningTotal	Ao longo de ShowAsAxis, o valor mais os valores dos membros anteriores com o mesmo pai, na ordem de exibição.
pvwDifferenceFrom	O valor menos o valor do membro base com o mesmo pai: o membro anterior quando ShowAsBaseItem está vazio, senão o membro com o rótulo ShowAsBaseItem (ignorando maiúsculas e minúsculas).
pvwRank	A posição entre os membros com o mesmo pai, o maior é 1. Os empates compartilham uma posição e a próxima posição é pulada: 1, 2, 2, 4.

- Um modo de porcentagem mostra 25 para 25 por cento, formatado como `0.00%`. Um valor vazio, uma base ausente e uma base 0 dão uma célula vazia.
- ShowAsAxis (pvxColumns por padrão) é o eixo ao longo do qual são tomados o total acumulado, a diferença e a posição.
- O primeiro membro de um grupo não tem com o que se comparar em pvwDifferenceFrom, e o próprio membro base fica vazio, como no Excel. O total geral de ShowAsAxis não tem irmãos, portanto permanece vazio nos três últimos modos.
- A mesma coluna de origem pode ser adicionada duas vezes, uma como valor e outra com ShowAs, para mostrar ambos.

Exemplo

```
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;
```

Veja também

- Agregações
- Edições

Campos calculados

Um campo calculado é uma coluna numérica calculada para cada linha de origem a partir de uma expressão. Os campos o usam por meio de FieldName exatamente como uma coluna de origem. É um recurso avançado, que não faz parte da edição Community.

Nível de linha

A expressão é avaliada por linha de origem e depois agregada, de modo que a soma de `Profit = [Revenue] - [Cost]` é a soma dos lucros. Isso não é a fórmula sobre os totais que o Excel calcula: a média de uma margem por linha não é a margem dos totais. Para uma razão de totais, mostre as duas medidas e calcule a razão em OnGetCellText, ou use mostrar valores como.

Sintaxe

Elemento	Sintaxe
Números	Decimais, com ponto e um expoente opcional: 12, 0.5, 1E3.
Campos	[Field Name] ou um nome simples de letras, dígitos e sublinhados, comparado com as colunas de origem e os outros campos calculados, ignorando maiúsculas e minúsculas.
Operadores	Da menor precedência para a maior: `OR` ; `AND` ; `NOT` ; `= <> < <= > >=` ; `+ -` ; `* /` ; `-` e `+` unários; `^` (potência, associativa à direita). Parênteses agrupam.
Funções	IF(c, a, b), ABS(x), ROUND(x, n), MIN(x, ...), MAX(x, ...), SQRT(x), ISNULL(x, v), YEAR(d), MONTH(d), sem diferenciar maiúsculas de minúsculas.

- Um operando vazio dá um resultado vazio, exceto em ISNULL, que retorna v quando x está vazio, e em IF, que olha apenas a condição e o ramo que toma.
- A divisão por zero, a raiz quadrada de um número negativo e um estouro dão um resultado vazio.
- As comparações e os operadores lógicos retornam 1 ou 0. Qualquer valor diferente de 0 é verdadeiro.
- ROUND arredonda metades para longe de zero, n vai de -15 a 15.

Erros

Um erro de sintaxe ou uma referência circular entre campos calculados gera EsgcPivotError no próximo recálculo, com o nome do campo calculado na mensagem. Um nome de campo que não foi carregado não é um erro: o campo calculado permanece vazio até que essa coluna chegue, de modo que o layout pode ser definido antes dos dados. Uma Expression vazia também dá valores vazios.

Exemplo

```
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;
```

Veja também

- TsgcPivotCalculatedField
- Agregações

Persistência do layout

O layout de uma tabela dinâmica, seus campos, filtros, totais e membros expandidos, é salvo e carregado como JSON. É um recurso avançado, que não faz parte da edição Community.

Método	Descrição
SaveLayoutToString: string	O layout como um texto JSON.
LoadLayoutFromString(aJSON)	Substitui o layout com um único recálculo.
SaveLayoutToFile(aFileName)	Grava o JSON como um arquivo UTF-8.
LoadLayoutFromFile(aFileName)	Lê um arquivo UTF-8 e substitui o layout.

TsgcPivotEngine e as duas grades têm esses métodos. As versões das grades também armazenam as larguras de coluna que o usuário alterou.

Formato

O documento é um único objeto: `v` (a versão do formato, 1), `rowGrandTotals`, `columnGrandTotals`, `subtotalPosition`, `calculatedFields` (nome e expressão), `fields` (um objeto por campo, agrupado por área, com todas as propriedades de TsgcPivotField), `expanded` (o estado de expansão dos níveis e dos caminhos de membro das linhas e das colunas) e, a partir de uma grade, `columnWidths`. As enumerações são gravadas como o nome do membro sem seu prefixo e com a primeira letra minúscula: pvfCountDistinct é `countDistinct`.

Regras de carga

- A carga é tolerante: membros desconhecidos são ignorados, e um membro ausente ou um valor de tipo errado mantém o padrão.
- Quando o engine já tem dados, um campo cujo nome não é uma coluna de origem nem um campo calculado é ignorado.
- Um layout carregado antes dos dados mantém seu estado de expansão até que os campos correspondam aos dados.
- Um JSON malformado, ou uma raiz que não é um objeto, gera EsgcPivotError e deixa o layout inalterado.

Exemplo

```
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;
```

Veja também

- Áreas
- Edições

Exportação

O engine e as duas grades exportam a visão atual, com seus totais e seu estado de expansão. Todo método tem uma sobrecarga para arquivo e para stream, e o texto é gravado em UTF-8 sem BOM.

Método	Saída
SaveToCSV	CSV RFC 4180 com registros CRLF. Uma coluna por nível de linha, uma linha de cabeçalho por nível de coluna mais a linha de medidas, e um rótulo de cabeçalho repetido em cada célula que ele cobre. Os valores são gravados como números invariantes, 1250.5, sem o formato de exibição, e uma célula vazia é um campo vazio.
SaveToHTML	Um documento HTML completo com uma tabela. Os cabeçalhos mesclados usam colspan e rowspan, e os valores são os textos formatados das células, com o EmptyText do engine em uma célula vazia.
SaveToJSON	O snapshot estruturado: os títulos dos níveis, as medidas com seus formatos, e os itens de coluna e de linha com seu tipo (member, total ou grandTotal), nível, rótulo, caminho e estado de expansão. Um item de linha também tem seus valores após mostrar valores como, null quando vazio.
SaveToPNG, SaveToSVG	Somente nas grades e no renderer: a grade inteira como imagem. Largura e altura em DIPs, 0 é o tamanho do conteúdo, até 16384 pixels.

Como tabela

TsgcPivotEngine e as duas grades implementam IsgcUITable por meio de ResultTable, de modo que o resultado alimenta qualquer consumidor do sgcUI, como uma origem de dados de gráfico do sgcCharts ou outra tabela dinâmica. As linhas são as folhas de linha visíveis, membros recolhidos incluídos e linhas de total excluídas. As colunas são uma coluna de texto por nível de linha, depois uma coluna por folha de coluna visível e medida, chamada *<caminho da coluna>* | *<título da medida>*, ou apenas o título da medida quando não há campos de coluna. Uma célula de valor vazia tem o EmptyText do engine como TableText e Null como TableValue. TableVersion muda a cada novo snapshot. Veja Interoperabilidade com gráficos.

Exemplo

```
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;
```

Veja também

- Início rápido headless
- TsgcPivotGridRenderer

Expandir e recolher

Todo membro que tem filhos, nas linhas ou nas colunas, pode ser expandido para mostrá-los ou recolhido em uma única linha ou coluna. Um membro recolhido mantém seu lugar e mostra os totais de tudo que está abaixo dele.

Nas grades

- Os botões + e - nos cabeçalhos de linha e de coluna alternam um membro. ShowExpandButtons os oculta.
- Um clique duplo em um cabeçalho de linha alterna seu membro.
- Com o cabeçalho de linha em foco, Right expande, Left recolhe e Enter alterna. As teclas + e - expandem e recolhem o membro da linha selecionada a partir de qualquer coluna. Veja Teclado e acessibilidade.
- OnExpandChange é gerado depois que o usuário expandiu ou recolheu um membro, com aAxis e aItem. Ele não é gerado pelos métodos abaixo.

No código

Membro	Efeito
TsgcPivotEngine.ExpandAll(aAxis), CollapseAll(aAxis)	Todos os membros de pvxRows ou pvxColumns.
TsgcPivotEngine.ExpandLevel(aAxis, aLevel), CollapseLevel(aAxis, aLevel)	Todos os membros de um nível, 0 é o nível mais externo.
TsgcPivotEngine.SetExpanded(aAxis, aItem, aValue), ToggleExpanded(aAxis, aItem)	Um item do snapshot atual, por seu índice em Result.Item. Um item sem filhos é deixado como está.
TsgcPivotField.Expanded	O estado inicial dos membros do nível desse campo. Padrão True. Veja Agrupamento de datas.
TsgcPivotEngine.GetExpandState(aAxis, aState), SetExpandState(aAxis, aState)	O estado de expansão explícito de um eixo como uma lista de strings, por exemplo para copiá-lo para outro engine. Edição Advanced.
TsgcPivotItem.Expandable, Expanded	Lidos de Result.Item: o item tem filhos, e eles estão visíveis.

Como o estado é mantido

- O estado é armazenado por caminho do membro, como North / Spain, e não por índice de item. Ele sobrevive a um recálculo, a uma recarga dos dados, a um filtro e a uma nova ordem de classificação, e um membro que desaparece e volta mantém seu estado.
- Quando os campos de um eixo mudam (outro campo, outra ordem ou outro agrupamento de datas), o estado desse eixo recomeça a partir da propriedade Expanded de seus campos.
- Expandir ou recolher constrói um novo snapshot sobre os dados calculados sem agregar novamente, portanto é rápido mesmo em um resultado grande. Veja a tabela de desempenho em Arquitetura.
- As exportações e ResultTable seguem o estado atual: um membro recolhido é uma linha com seus totais.
- Persistência do layout salva e carrega o estado de expansão junto com o layout.

Exemplo

```
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 é chamado após Unlock, porque a alteração publica um novo snapshot.

Veja também

- Totais
- Persistência do layout
- TsgcPivotEngine

Ordenação

Todo campo de linha e de coluna ordena seus membros dentro de cada pai, de modo que os países de cada região são ordenados separadamente. Três propriedades de TsgcPivotField a controlam.

Propriedade	Efeito
SortBy	pvsLabel (padrão) ordena pelo rótulo do membro. pvsValue ordena pelo total de cada membro.
SortOrder	pvoAscending (padrão) ou pvoDescending.
SortMeasure	O campo de dados, por seu AreaIndex, cujos totais uma ordenação por valor compara. Padrão 0. O Top N também o usa.

Regras

- Uma ordenação por rótulo compara membros de texto ignorando maiúsculas e minúsculas, com as regras da localidade, e membros numéricos e de data pelo seu valor, de modo que 9 vem antes de 10 e 2024-10 depois de 2024-09.
- Uma ordenação por valor compara o total de cada membro para a medida SortMeasure: o total sobre todas as colunas para um campo de linha, e sobre todas as linhas para um campo de coluna. Os membros com total vazio ficam por último, e os membros com o mesmo total mantêm a ordem de rótulo.
- Quando SortMeasure não é o AreaIndex de um campo de dados, os membros são ordenados por rótulo.
- O membro em branco e o membro Others do Top N ficam no final nas duas ordens, Others depois do membro em branco.
- O Top N escolhe primeiro os membros com os maiores totais de SortMeasure, depois SortBy e SortOrder os colocam em ordem.
- Uma alteração de SortBy, SortOrder ou SortMeasure recalcula a tabela dinâmica, uma vez por bloco BeginUpdate e EndUpdate.
- As grades não ordenam quando um cabeçalho é clicado. Para permitir que o usuário ordene, defina as propriedades no código, por exemplo em OnCellClick quando o Part acertado é pvhColumnHeader.

Exemplo

```
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;
```

Veja também

- Top N
- Agrupamento de datas
- TsgcPivotField

Drill-down

Um drill-down lê as linhas de origem por trás de uma célula: as linhas que foram agregadas em seu valor. O engine as retorna como números de linha ou como uma tabela.

Método	Retorna	Descrição
TsgcPivotEngine.GetSourceRows(aRowItem, aDataCol, var aRows)	Integer	Os números das linhas de origem, em ordem de linha, e sua contagem. aRowItem é um item de linha de Result e aDataCol uma coluna de dados. aDataCol -1 usa todas as colunas do item de linha.
TsgcPivotEngine.DrillDown(aRowItem, aDataCol)	TsgcPivotSourceView	As mesmas linhas como uma IsgcUITable com todas as colunas do armazenamento, inclusive as colunas que não estão no layout. Quem chama a libera.
TsgcPivotEngine.SourceValue(aRow, aCol)	Variant	Um valor de uma linha de origem, Null quando vazio. SourceColumnName fornece os nomes das colunas.

Quais linhas

- Uma célula de membro retorna as linhas do seu membro de linha e do seu membro de coluna. Um subtotal ou um total geral retorna todas as linhas abaixo dele, e o membro Others do Top N retorna as linhas de todos os membros que ele contém.
- As linhas excluídas por um filtro de membro, um filtro de rótulo, um filtro de relatório ou OnFilterRow nunca são retornadas.
- Todas as medidas de um item de coluna têm as mesmas linhas, portanto a medida de aDataCol não altera o resultado.

Nas grades

Enter em uma célula de dados e um clique duplo em uma célula de dados disparam OnDrillDown com aRowItem, aDataCol e var aHandled. A grade não tem ação própria, portanto o aplicativo mostra as linhas em um diálogo, em uma grade ou em um relatório. A demo DataSetPivot abre os pedidos por trás de uma célula.

Tempo de vida

- TsgcPivotSourceView lê o armazenamento do engine ao vivo. Depois que os dados de origem mudam, ele fica sem linhas, e deve ser liberado antes do engine.
- Os índices de item pertencem ao snapshot atual. Obtenha-os dos eventos da grade, ou leia Result entre Lock e Unlock.

Exemplo

```
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;
```

Veja também

- TsgcPivotEngine
- TsgcVCLPivotGrid
- Filtros

Formatação condicional

A coleção FormatRules das grades e do renderer colore as células de dados de acordo com seus valores. As mesmas regras valem na tela e nas imagens PNG e SVG, porque as grades e as imagens são desenhadas pelo mesmo renderer. Em tempo de design, as regras são editadas com o editor de coleções do Object Inspector.

Kind	O que ela faz
pvrValue	Uma regra de valor. Quando o valor satisfaz Operator, Value e Value2 (os operadores dos filtros), a célula recebe BackColor, ForeColor e Bold. pvcNone satisfaz todo valor que não é vazio.
pvrColorScale	Uma escala de cores. As células recebem uma cor entre MinColor e MaxColor, segundo sua posição entre o menor e o maior valor das células de membro da medida. Operator, Value e Value2 selecionam os valores que ela colore.

Quais células

- Measure seleciona o campo de dados por seu AreaIndex, e -1 aplica a regra a todas as medidas.
- Scope seleciona os tipos de célula: pveCells (o padrão, as células de membro), pveTotals e pveGrandTotals.
- Uma célula vazia não satisfaz nenhuma regra e nunca é colorida. Ela mostra o EmptyText do engine.

Ordem das regras

- As regras são executadas na ordem da coleção. Toda regra habilitada que satisfaz aplica suas cores diferentes de 0, de modo que uma correspondência posterior substitui uma anterior, e Bold se acumula.
- StopOnMatch encerra a avaliação para aquela célula, de modo que a primeira regra que satisfaz vence. O exemplo o usa para três faixas.
- OnGetCellStyle é executado após as regras com var aBack, aFore e aBold, e pode alterar qualquer um deles. Dentro dele, Renderer.CurrentResult é o snapshot que está sendo desenhado.

Cores

As cores são valores TsgcUIColor no formato $AARRGGBB, de modo que $FF2E7D32 é um verde opaco, e 0 mantém a cor do tema. Com o tema escuro, escolha cores que mantenham o texto legível, e combine a cor com Bold para que o significado não dependa apenas da cor.

Exemplo

```
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;
```

Veja também

- TsgcPivotFormatRule
- TsgcPivotGridRenderer
- Teclado e acessibilidade

Interoperabilidade com gráficos

O sgcPivotGrid troca dados com os outros produtos sgcUI por meio de IsgcUITable, a interface de tabela do sgcUI-Core na unit sgcUICore_Table. Os pacotes do sgcPivotGrid não dependem de sgcCharts nem de sgcGrid, o aplicativo os conecta.

Da tabela dinâmica para o sgcCharts

TsgcPivotEngine e as duas grades implementam IsgcUITable por meio de ResultTable. Defina a propriedade Table de um TsgcChartDataSource_Table do sgcCharts como a grade dinâmica, ou chame seu SetTable com um engine sem grade.

Parte da tabela	Conteúdo
Linhas	As folhas de linha visíveis: membros recolhidos incluídos, linhas de subtotal e de total geral excluídas.
Colunas de texto	Uma por nível de linha, com o nome do título do nível, com o rótulo do membro nesse nível.
Colunas de valor	Uma por folha de coluna visível e medida, chamada *<caminho da coluna> | <título da medida>*, por exemplo `Online | Sales` , ou apenas o título da medida quando não há campos de coluna.
TableValue	O rótulo de uma coluna de texto, e o valor exibido de uma coluna de valor como Double (após mostrar valores como), ou Null quando a célula está vazia.
TableText	O rótulo, ou o texto formatado da célula. Uma célula vazia dá o EmptyText do engine.

- Mapeie as colunas com a origem de dados: YColumn nomeia a coluna de valor, LabelColumn o nível de linha de um gráfico de pizza, funil ou treemap (vários níveis separados por ;) e XColumn os valores X de uma série XY.
- Dê ao campo de dados um Caption, como Sales, para que os nomes das colunas não mudem com a agregação.
- Cada novo snapshot altera TableVersion e notifica os ouvintes, de modo que o gráfico recarrega após um recálculo, uma expansão ou um recolhimento.
- A demo GridToPivotToChart desenha barras agrupadas com uma origem de dados por canal e preenche as categorias do eixo X a partir dos rótulos de linha em OnCalculated.

Do sgcGrid para a tabela dinâmica

TsgcPivotSource_Table carrega qualquer IsgcUITable no engine: um TsgcVCLGrid, um TsgcFMXGrid ou um TsgcGridData do sgcGrid. Ele recarrega quando TableVersion muda, de modo que uma edição na grade atualiza a tabela dinâmica e depois o gráfico. Agrupe muitas alterações da grade em BeginUpdate e EndUpdate dos dados da grade, o que resulta em uma única recarga.

De uma tabela dinâmica para outra

O resultado de uma tabela dinâmica também é uma tabela, portanto TsgcPivotSource_Table pode carregá-lo em uma segunda tabela dinâmica.

Lendo a tabela no código

Leia a tabela entre TableLock e TableUnlock. A contagem de referências está desabilitada, o engine é o dono da tabela, portanto descarte todas as referências de interface antes de liberar o engine.

Exemplo

```
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;
```

Veja também

- Exportação
- TsgcPivotSource_Table
- Demos

Teclado e acessibilidade

As duas grades podem ser usadas apenas com o teclado. As mesmas teclas funcionam em TsgcVCLPivotGrid e TsgcFMXPivotGrid, onde Cmd no macOS funciona como Ctrl.

Tecla	Ação
Up, Down	A linha anterior ou a seguinte.
Page Up, Page Down	Uma página para cima ou para baixo.
Home, End	A primeira ou a última coluna de dados da linha. Com Ctrl, a primeira ou a última célula da grade.
Left	A coluna de dados anterior. Da primeira coluna de dados vai para o cabeçalho de linha e, no cabeçalho de linha, recolhe o membro.
Right	No cabeçalho de linha expande um membro recolhido, caso contrário vai para a primeira coluna de dados. Nos dados vai para a coluna seguinte.
+, -	No teclado numérico ou no teclado principal: expande ou recolhe o membro da linha selecionada, a partir de qualquer coluna.
Enter	Em uma célula de dados dispara OnDrillDown. No cabeçalho de linha alterna o membro.

- Quando nada está selecionado, a primeira tecla de navegação seleciona a primeira célula.
- TsgcVCLPivotGrid tem TabStop True por padrão e TsgcFMXPivotGrid tem CanFocus True. Um clique dá o foco à grade.
- No código, SelectedRow, SelectedCol e ScrollToCell movem a seleção, e OnSelectionChange informa cada alteração.
- As teclas expandem e recolhem membros de linha. Os membros de coluna se expandem com os botões dos cabeçalhos de coluna ou no código, veja Expandir e recolher.
- As listas de campos são usadas com o mouse ou por toque. Escape cancela um arraste, e TsgcFMXPivotFieldList também rola sua lista com Up, Down, Page Up, Page Down, Home e End. Para usuários de teclado, ofereça os layouts no código com AddField e MoveField, ou como layouts salvos.

Acessibilidade

- As grades e as listas de campos são desenhadas pelo sgcUICore e não publicam seu conteúdo para leitores de tela por meio de UI Automation ou MSAA. Para usuários de leitores de tela, ofereça os dados como HTML: SaveToHTML grava uma tabela real com células de cabeçalho, colspan e rowspan. ResultTable também pode preencher um controle padrão.
- Os tamanhos estão em DIPs e acompanham o DPI do monitor. FontSize, RowHeight e HeaderHeight deixam o texto e as linhas maiores.
- Theme seleciona o tema claro ou escuro, e AccentColor substitui a cor de destaque do tema.
- Formatação condicional pode adicionar Bold a uma cor, para que um valor não dependa apenas da cor.

Exemplo

```
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;
```

Veja também

- TsgcVCLPivotGrid
- Drill-down

Desempenho

O engine foi feito para grandes volumes de dados: um armazenamento de colunas tipado, uma única passagem sobre as linhas de origem por recálculo e desenho virtual. O grupo de desempenho do QA mede os principais cenários em relação aos orçamentos listados em Arquitetura. Estas diretrizes mantêm um aplicativo dentro deles.

Carga

- Carregue dentro de BeginLoad e EndLoad. O engine recalcula uma vez, no EndLoad mais externo. Fora de um bloco, cada AppendRow e DefineColumn recalcula imediatamente.
- Defina as colunas com DefineColumn(aName, aType). Sem tipo, o primeiro valor que não seja vazio o define, e uma coluna inteira se torna uma coluna de ponto flutuante quando chega um valor fracionário.
- TsgcPivotSource_DataSet recarrega o dataset inteiro após cada post enquanto LiveUpdates é True. Para um lote de edições, defina LiveUpdates ou Active como False e chame Load no final.
- TsgcPivotSource_Table recarrega somente quando TableVersion muda. Agrupe as alterações de uma grade em seu BeginUpdate e EndUpdate.

Layout e visão

- Altere o layout dentro de BeginUpdate e EndUpdate para um único recálculo. Para lotes mais longos, defina AutoRecalculate como False e chame Recalculate no final.
- DeferUpdate das listas de campos mantém as alterações do usuário pendentes até ApplyChanges.
- Expandir, recolher, RowGrandTotals, ColumnGrandTotals, SubtotalPosition, RowLayout e EmptyText alteram somente a visão. O engine constrói um novo snapshot sobre os dados calculados sem agregar novamente.

Tamanho do resultado

- O número de células cresce com os membros das linhas vezes os membros das colunas. Um campo com muitos membros, como um número de pedido ou uma data por dia, nos dois eixos os multiplica. Um layout que precise de mais de MaxCells células (5.000.000 por padrão) gera EsgcPivotError. Agrupe as datas, mova o campo para os filtros ou use Top N.
- Cada campo de dados adiciona um valor por célula. CountDistinct guarda os valores distintos de cada célula, e StdDev e Var guardam mais estado que Sum, portanto custam mais memória e tempo em layouts grandes.

Eventos

- OnFilterRow é executado para cada linha de origem em cada recálculo. Encontre os índices das colunas uma vez, fora do evento, como faz o exemplo de Filtros. Os filtros de membro e de rótulo são decididos uma vez por membro, portanto custam menos.
- OnGetCellText e OnGetCellStyle são executados para cada célula visível em cada quadro. Mantenha-os curtos.
- Mantenha o bloqueio de Result apenas enquanto o lê.

Exemplo

```
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;
```

Veja também

- Arquitetura
- TsgcPivotEngine
- Solução de problemas

TsgcPivotGridAIQuery

Unit: sgcPivotGridWS_AI
Package: sgcPivotGridWS (Delphi) e sgcPivotGridWSC (C++Builder), a ponte sgcWebSockets do sgcPivotGrid. Edição Registered, somente Windows (Win32 e Win64).
TsgcPivotGridAIQuery é um componente não visual que transforma uma pergunta em linguagem natural, como *top 3 countries by sales*, no layout de uma tabela dinâmica. Ele envia a pergunta e um catálogo dos campos de origem a um provedor de IA por meio de um **TsgcAIChat** do sgcWebSockets, valida a resposta JSON em relação aos campos do motor e a aplica em um único recálculo. As linhas de origem nunca são enviadas. Veja Pergunte à sua tabela dinâmica para saber o que é enviado, como a resposta é validada e o modo assíncrono.
Solte um TsgcAIChat e um TsgcPivotGridAIQuery da página **SGC PivotGrid** da paleta, defina Chat e PivotGrid e chame Ask. O provedor, o modelo e a chave de API são propriedades do chat: Provider, ChatOptions.Model e ChatOptions.ApiKey.

Propriedades

Propriedade	Tipo	Descrição
Active	Boolean	Com False, Ask e AskAsync retornam False e não fazem nada. True por padrão.
Chat	TsgcAI_Chat	O chat que conversa com o provedor: um TsgcAIChat ou qualquer descendente de TsgcAI_Chat. Sem chat, Ask dispara OnError.
PivotGrid	TComponent	A tabela dinâmica cujo layout é alterado: um TsgcVCLPivotGrid, um TsgcFMXPivotGrid ou qualquer componente que implemente IsgcPivotEngineHost.
Engine	TsgcPivotEngine	Public. Um motor para usar sem grade, por exemplo em um servidor. Tem prioridade sobre PivotGrid.
SendSampleValues	Boolean	Envia com o catálogo alguns valores de cada campo de texto, inteiro, data e booleano, para que o provedor possa reconhecer os membros de uma pergunta como *online sales*. True por padrão.
MaxSampleValues	Integer	O número de valores de amostra por campo, limitado a 0..5. 3 por padrão.
Language	string	O idioma da explicação, por exemplo `Spanish` . Vazio, o padrão, a escreve no idioma da pergunta.
ReplaceLayout	Boolean	True, o padrão, pede um novo layout. False envia também o layout atual, de modo que uma pergunta como *add the channel to the columns* o altera. A resposta é o novo layout completo nos dois casos.
Busy	Boolean	Public, somente leitura. True enquanto uma requisição de AskAsync está em execução.
LastLayout	TsgcPivotAILayout	Public. O layout aplicado pela última resposta. Fica vazio depois de Undo.
LastExplanation	string	Public. A explicação da última resposta aplicada, uma frase curta para uma barra de status.

Eventos

Evento	Tipo	Descrição
OnBeforeAsk	TsgcPivotAIBeforeAskEvent	`(Sender; var aPrompt: string; var aAllow: Boolean)`. Dispara na thread principal antes da requisição, com a mensagem do usuário. Altere aPrompt para acrescentar suas próprias instruções, ou defina aAllow como False para cancelar a pergunta em silêncio.
OnLayout	TsgcPivotAILayoutEvent	`(Sender; aLayout: TsgcPivotAILayout; var aAccept: Boolean)`. Dispara com o layout validado antes de ele ser aplicado. Inspecione ou edite aLayout, ou defina aAccept como False para manter o layout atual.
OnApplied	TNotifyEvent	Dispara depois que o layout foi aplicado. LastLayout e LastExplanation contêm a resposta.
OnError	TsgcPivotAIErrorEvent	`(Sender; const aError: string)`. Dispara quando nada foi aplicado: sem chat, sem motor, uma pergunta vazia, uma requisição ainda em execução, um erro do provedor, uma resposta sem objeto JSON ou uma resposta sem nenhum campo de linha, coluna ou dados utilizável. O layout permanece como estava.

Métodos

Método	Retorna	Descrição
Ask(aQuestion)	Boolean	Consulta o provedor, valida a resposta e a aplica. Síncrono, somente na thread principal. True quando um layout foi aplicado.
AskAsync(aQuestion)	Boolean	O mesmo, com a requisição em uma thread de trabalho. A resposta é validada e aplicada na thread principal, e então OnApplied ou OnError dispara. False quando Busy é True ou a requisição não iniciou.
Undo		Restaura o layout anterior à última resposta aplicada. Apenas um nível é mantido.
BuildPrompt(aQuestion)	string	A mensagem do usuário que Ask envia para aQuestion: a pergunta, a linha do idioma e o catálogo de campos. Vazia quando não há motor. Use-a para mostrar ao usuário o que sai da aplicação.
ValidateLayoutJSON(aJSON, aLayout)	Boolean	Extrai, analisa e valida uma resposta ou um layout salvo em relação aos campos do motor, gravando em aLayout. False, com aLayout vazio, quando nenhum objeto JSON é encontrado ou nenhum campo de linha, coluna ou dados sobra.
ResolveEngine	TsgcPivotEngine	Engine quando atribuído, senão o motor de PivotGrid, senão nil.

Os descendentes podem sobrescrever três métodos virtuais protegidos: **DoAskProvider**, o único método que conversa com o provedor (um teste pode retornar daqui uma resposta predefinida, como a demo AskYourPivot faz em seu autoteste), **DoBuildFields**, que constrói o catálogo de campos (um campo omitido não pode ser enviado nem aceito), e **DoApply**, que substitui o layout do motor.

TsgcPivotAILayout

O layout de uma resposta, após a validação. OnLayout o recebe e LastLayout guarda o aplicado.

Membro	Descrição
Rows, Columns	Os campos de eixo em ordem: FieldName e Grouping (TsgcPivotDateGrouping).
Data	Os campos de dados em ordem: FieldName e Aggregation (TsgcPivotAggregation). O índice de uma entrada é o índice da medida a que se referem filtros, ordenações e Top N.
Filters	Tipo `pafMembers` (Members, Exclude), `pafLabel` (FilterOperator, Value, Value2) ou `pafValue` (FilterOperator, NumValue, NumValue2, Measure).
Sorts	FieldName, SortBy, SortOrder e Measure.
TopN	FieldName, Count, Measure e Others.
Explanation	A explicação do provedor, limpa.
Clear, IsEmpty, Assign	Esvazia o layout, testa a ausência de entradas de linha, coluna e dados, copia outro layout.
ToJSON	O layout como um único objeto JSON compacto no formato do esquema, de modo que um layout salvo seja validado de novo.
ToText	Um resumo de uma linha do layout, para uma barra de status.

Exemplo

Um auxiliar que possui o chat e a consulta, pergunta de forma síncrona e recusa um layout com mais de dois campos de linha:

```
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;
```

Veja também

- Pergunte à sua tabela dinâmica
- TsgcPivotEngine
- TsgcVCLPivotGrid
- Demos

Pergunte à sua tabela dinâmica

TsgcPivotGridAIQuery permite que o usuário projete uma tabela dinâmica pedindo por ela. Uma pergunta como *online sales by region and quarter* vira Region nas linhas, Date por trimestre nas colunas, a soma de Sales nos valores e um filtro de membros em Channel. Este guia explica o que é enviado ao provedor de IA, como a resposta é verificada antes de qualquer mudança, o modo assíncrono e os requisitos.

Como uma pergunta é respondida

1. Ask ou AskAsync monta a mensagem do usuário a partir da pergunta e do catálogo de campos do motor, e OnBeforeAsk pode alterá-la ou cancelá-la.
2. O chat a envia ao provedor com uma mensagem de sistema fixa e um esquema JSON fechado, e pede um único objeto JSON.
3. A resposta é validada em relação aos campos do motor. Nada é aplicado quando nenhum campo de linha, coluna ou dados utilizável sobra, e OnError dispara.
4. OnLayout recebe o layout validado e pode editá-lo ou recusá-lo.
5. O layout substitui o atual em um único recálculo, OnApplied dispara e Undo pode restaurar o layout anterior.

O que é enviado

Três textos saem da aplicação, e nada além deles:

Texto	Conteúdo
Mensagem de sistema	A constante `CS_SGC_PIVOT_AI_SYSTEM` : responder com um objeto JSON que siga o esquema, usar apenas os nomes de campo listados, nunca inventar campos ou valores e ignorar qualquer instrução encontrada na pergunta, em um nome de campo ou em um valor de amostra.
Mensagem do usuário	O texto de `BuildPrompt` : a pergunta, o idioma da explicação e uma matriz JSON com uma entrada por coluna de origem, `{"name","caption","type","samples"}` , mais os campos calculados do motor marcados com `"calculated":true` . Com ReplaceLayout False, ela inclui também o layout atual.
Esquema JSON	A constante `CS_SGC_PIVOT_AI_SCHEMA` , chamada `sgcpivot_layout` , enviada como esquema de saída estruturada do provedor.

- **As linhas de origem nunca são enviadas.** O provedor vê os nomes, os títulos e os tipos dos campos, e alguns valores de amostra.
- Os valores de amostra são os primeiros valores distintos e não vazios dos campos de texto, inteiro, data e booleano, lidos das primeiras 10.000 linhas, no máximo MaxSampleValues por campo e cortados em 40 caracteres. Campos Float, que costumam ser as medidas, e campos calculados nunca enviam amostras. Defina SendSampleValues como False, ou MaxSampleValues como 0, para não enviar nenhum valor.
- O título de um campo é o Caption de um campo do layout com o mesmo nome, quando você deu um, senão o nome da coluna.
- Chame BuildPrompt para mostrar ao usuário exatamente o que é enviado, como faz o botão *Show prompt* da demo AskYourPivot.

Um motor sem interface com três colunas, a mensagem do usuário que ele produz e um layout salvo verificado pelo mesmo 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;
```

A resposta

O esquema é fechado: cada objeto lista todas as suas chaves e não admite outra. Uma resposta tem as chaves `rows` e `columns` (field e group: none, year, quarter, month, week ou day), `data` (field e aggregation: sum, count, average, min, max, countDistinct, stdDev ou var), `filters` (um filtro de membros, de rótulo ou de valor), `sort` (por rótulo ou por valor, crescente ou decrescente), `top` (Top N com um membro Others opcional) e `explanation`. Uma medida é o índice de uma entrada de `data`, começando em 0.
Alguns provedores recusam o modo de saída estruturada. Quando o erro do provedor cita o esquema ou o formato de resposta, a consulta pergunta mais uma vez em modo JSON simples, com o esquema colado na mensagem do usuário. As quatro configurações do chat que ela altera (ResponseFormat, JSONSchema, JSONSchemaName e SystemMessage) são restauradas e o histórico do chat é limpo após cada pergunta, de modo que o chat pode ser compartilhado com outro código.

Validação

A resposta é uma entrada não confiável. ValidateLayoutJSON é a fronteira entre o provedor e o motor, e toda resposta passa por ele antes de OnLayout:

- A resposta é cortada em 64 KB. Procura-se primeiro o primeiro bloco de código delimitado, depois o primeiro objeto JSON balanceado, ignorando as chaves dentro de strings.

- O nome de um campo deve corresponder exatamente a um campo do motor, sem diferenciar maiúsculas de minúsculas e ignorando os espaços externos. Campos calculados são aceitos apenas nos valores. Campos, chaves e valores de enumeração desconhecidos são descartados em silêncio.
- Uma agregação numérica de um campo de texto ou booleano vira Count, e a explicação informa isso. Um agrupamento de datas só é mantido para campos de data.
- No máximo 4 entradas de linha, 4 de coluna, 6 de dados, 8 de filtro, 4 de ordenação e 2 de Top N, 50 membros por filtro e N entre 1 e 100.
- Caracteres de controle são removidos, os textos cortados em 200 caracteres e a explicação em 300, sem colchetes angulares.
- Um campo nas linhas ou nas colunas não é colocado também na área de filtros. Um filtro de membros de um campo que não está em nenhum eixo vai para a área de filtros.
- Uma resposta com campos de eixo mas sem dados recebe o Count do primeiro campo de linha. Uma resposta sem nenhum campo de linha, coluna ou dados é rejeitada por inteiro: nada muda e OnError dispara.

Como o mesmo método valida um layout salvo, uma aplicação pode guardar LastLayout.ToJSON e verificá-lo de novo mais tarde em relação aos campos atuais.

Aplicação e Undo

Um layout validado substitui o layout do motor em um único recálculo: as linhas, as colunas e os campos de dados em ordem, depois os filtros, as ordenações e Top N no primeiro campo com esse nome nas linhas, senão nas colunas. Undo restaura o layout anterior à última resposta aplicada, e LastLayout fica então vazio.

Perguntas assíncronas

Um provedor pode levar vários segundos para responder. AskAsync executa apenas a requisição em uma thread de trabalho, de modo que a janela continua responsiva. OnBeforeAsk dispara na thread principal antes de a requisição começar, e a resposta é validada e aplicada na thread principal por meio de Synchronize, portanto OnLayout, OnApplied e OnError também disparam ali.

- Uma pergunta por vez: AskAsync retorna False enquanto Busy é True, e OnError informa.
- Não use o chat a partir do seu próprio código enquanto Busy for True.
- Liberar a consulta enquanto uma requisição está em execução espera a thread de trabalho e descarta a resposta.

```
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;
```

Edição e requisitos

Requisito	Detalhes
Edição	Somente Registered. A ponte sgcPivotGridWS não faz parte da edição Community.
sgcWebSockets	A ponte conversa com o provedor por meio do TsgcAIChat do sgcWebSockets (unit sgcAI), portanto o sgcWebSockets com seus componentes de IA precisa estar instalado. Os provedores são os do TsgcAIChat: OpenAI, Anthropic, Gemini, DeepSeek, Ollama, Grok e Mistral.
Plataformas	Somente Windows, Win32 e Win64, Delphi 7 a 13 e as versões correspondentes do C++Builder.
Pacotes	sgcPivotGridWS<V> e dclsgcPivotGridWS<V> para Delphi, sgcPivotGridWSC<V> e dclsgcPivotGridWSC<V> para C++Builder. Eles exigem sgcPivotGrid, sgcUICore e sgcWebSockets.
Chave de API	Cada provedor precisa da sua própria chave, definida em ChatOptions.ApiKey do chat. O Ollama roda localmente e não precisa de nenhuma. Não guarde a chave em um arquivo de formulário.

TsgcAIChat só existe quando o sgcWebSockets é compilado com `SGC_AI`. Os pacotes que o instalador do sgcWebSockets compila o definem. Se você compilar os fontes do sgcWebSockets por conta própria e o compilador relatar TsgcAI_Chat como identificador não declarado em sgcPivotGridWS_AI, adicione `SGC_CUSTOM_INDY` às definições condicionais dessa compilação: `SGC_AI` depende dele.

A demo AskYourPivot

`Demos\PivotGrid\Delphi\AskYourPivot` coloca um TsgcVCLPivotGrid com 100.000 linhas de vendas geradas sob uma caixa de pergunta. Escolha o provedor, confira o modelo e digite sua chave de API, que fica apenas na memória. Ask usa AskAsync, Undo restaura o layout anterior, Examples traz quatro perguntas de exemplo e Show prompt abre a mensagem de sistema, a mensagem do usuário e o esquema. Execute-a com `--test` para verificá-la com respostas predefinidas, sem rede e sem chave. Veja Demos.

Veja também

- TsgcPivotGridAIQuery
- Áreas
- Filtros
- Top N

Visão geral do .NET

sgcPivotGrid .NET é um port gerenciado em C# da biblioteca Delphi sgcPivotGrid, não um wrapper em torno dela. As classes ficam no namespace **esegece.sgcPivotGrid** e mantêm os nomes do Delphi, de modo que **TsgcPivotEngine**, **TsgcPivotField**, **TsgcPivotResult**, **TsgcPivotGridRenderer**, **TsgcPivotFormatRule**, **TsgcPivotSource_DataSet** e **TsgcPivotSource_Table** se leem igual nas duas linguagens. As views seguem o padrão de nomes do Delphi: **TsgcWinFormsPivotGrid** e **TsgcWinFormsPivotFieldList** para WinForms, **TsgcWpfPivotGrid** e **TsgcWpfPivotFieldList** para WPF, e **TsgcAvaloniaPivotGrid** e **TsgcAvaloniaPivotFieldList** para Avalonia. Todas desenham com o port .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe, de modo que um pivot tem a mesma aparência em cada host e na saída PNG e SVG sem interface.
O motor e sua aritmética são os do Delphi, operação por operação: o armazenamento de colunas tipado, as oito agregações, o agrupamento de datas, os subtotais e totais gerais, os filtros de rótulo e de valor, a ordenação, expandir e recolher, o drill-down até as linhas de origem, as regras de formatação condicional e a exportação para CSV, JSON e HTML. A saída SVG do renderizador é idêntica byte a byte à do Delphi para a mesma cena, de modo que um relatório gerado em um servidor Linux coincide com o que um aplicativo desktop Delphi desenha.

Pacotes

O port é dividido em pacotes NuGet, de modo que um projeto inclui apenas o que usa.

PackageId	Contém	Frameworks de destino
esegece.sgcPivotGrid	O motor (**TsgcPivotEngine**) com seu armazenamento de colunas, campos, agregações, agrupamento de datas, totais, filtros, ordenação, drill-down e instantâneo (**TsgcPivotResult**), as regras de formato, o renderizador **TsgcPivotGridRenderer** com a saída SVG, o modelo da lista de campos, as duas origens e a exportação para CSV, JSON e HTML. Na edição Registered também os campos calculados, mostrar valores como, Top N e o JSON de layout. Sem dependência nativa.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPivotGrid.Rendering	A saída PNG (**SaveToPNG**), por meio do SkiaSharp.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPivotGrid.WinForms	**TsgcWinFormsPivotGrid**, **TsgcWinFormsPivotFieldList** e seu popup de membros.	net472, net8.0-windows, net9.0-windows
esegece.sgcPivotGrid.Wpf	**TsgcWpfPivotGrid**, **TsgcWpfPivotFieldList** e seu popup de membros.	net472, net8.0-windows, net9.0-windows
esegece.sgcPivotGrid.Avalonia	**TsgcAvaloniaPivotGrid**, **TsgcAvaloniaPivotFieldList** e seu popup de membros, no Avalonia 11.3.	net8.0, net9.0

Cada pacote de host depende de esegece.sgcPivotGrid.Rendering e do pacote sgcUICore do mesmo host, portanto basta adicionar o pacote do host.

Edições

Edição	IDs dos pacotes	Inclui
Registered	esegece.sgcPivotGrid, .Rendering, .WinForms, .Wpf e .Avalonia	Tudo: as três grades e listas de campos, o motor, ambas as origens, todas as agregações, o agrupamento de datas, os totais, os filtros, a ordenação, o drill-down, as regras de formato, a exportação, a saída sem interface e os quatro recursos avançados: campos calculados, mostrar valores como, Top N com Others e persistência do layout.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcPivotGrid.WinForms.Community.	Tudo, exceto os quatro recursos avançados: **CalculatedFields**, as propriedades **ShowAs**, **ShowAsAxis** e **ShowAsBaseItem**, **TopN**, **TopNOthers** e **OthersCaption**, os métodos **SaveLayout** e **LoadLayout**, e **GetExpandState** e **SetExpandState**.

É a mesma divisão das edições Delphi, veja Edições.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	TsgcVCLPivotGrid e TsgcVCLPivotFieldList	Windows, no .NET Framework 4.7.2 e no .NET 8 e 9.
WPF	TsgcFMXPivotGrid e TsgcFMXPivotFieldList	Windows, no .NET Framework 4.7.2 e no .NET 8 e 9.
Avalonia	TsgcFMXPivotGrid e TsgcFMXPivotFieldList	Windows, Linux e macOS, no .NET 8 e 9.
Sem interface	TsgcPivotGridRenderer	Qualquer plataforma do .NET Standard 2.0, incluindo servidores Linux.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Vinculação de dados no .NET
- Renderização sem interface no .NET
- Demos do .NET

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host; o núcleo e o pacote de renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcPivotGrid.WinForms
dotnet add package esegece.sgcPivotGrid.Wpf
dotnet add package esegece.sgcPivotGrid.Avalonia
```

Um servidor ou um aplicativo de console sem janela adiciona o núcleo e, para PNG, o pacote de renderização:

```
dotnet add package esegece.sgcPivotGrid
dotnet add package esegece.sgcPivotGrid.Rendering
```

Edição Community

```
dotnet add package esegece.sgcPivotGrid.WinForms.Community
dotnet add package esegece.sgcPivotGrid.Wpf.Community
dotnet add package esegece.sgcPivotGrid.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcPivotGrid.WinForms" Version="2026.10.0" />
</ItemGroup>
```

Referencie em um projeto os pacotes Registered ou os Community, nunca ambos: eles contêm os mesmos tipos no mesmo namespace. O código que usa um recurso avançado não compila com os pacotes Community; proteja-o com `#if !SGC_COMMUNITY` quando um projeto compila as duas edições, como faz o exemplo de Diferenças do .NET em relação ao Delphi.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:pivot="clr-namespace:esegece.sgcPivotGrid;assembly=esegece.sgcPivotGrid.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicativo desktop Avalonia. Mapeie o namespace no XAML com `xmlns:pivot="using:esegece.sgcPivotGrid"`.
- Sem interface: esegece.sgcPivotGrid sozinho basta para o motor, ambas as origens, CSV, JSON, HTML e SVG. Adicione esegece.sgcPivotGrid.Rendering para PNG; ele traz o SkiaSharp e seus recursos nativos.

Em C# cada host precisa de `using esegece.sgcPivotGrid;`, e de `using esegece.sgcUICore;` para **TsgcUITheme**, **sgcUICore_Theme** e os demais tipos dos temas.

Veja também

- Visão geral do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Renderização sem interface no .NET
- Demos do .NET

Início rápido WinForms

Este exemplo cria um formulário WinForms com uma grade pivot e sua lista de campos sobre 50.000 linhas de vendas geradas: Region e Product nas linhas, os anos da coluna Date nas colunas e a soma das vendas nas células. Uma caixa de seleção alterna a grade e a lista de campos entre o tema claro e o escuro, e um clique duplo em uma célula lê as linhas de origem por trás dela.

Passos

- Crie um projeto WinForms `net8.0-windows` e adicione o pacote **esegece.sgcPivotGrid.WinForms**, veja Instalação pelo NuGet.
- Crie um **TsgcWinFormsPivotGrid** e ancore-o. Sua propriedade **Pivot** é o **TsgcPivotEngine** que guarda as linhas e o layout.
- Carregue as linhas entre **BeginLoad** e **EndLoad**: **DefineColumn** dá nome a cada coluna e fixa seu tipo, **AppendRow** adiciona os valores na ordem das colunas. O motor recalcula uma única vez, no EndLoad mais externo.
- Monte o layout entre **BeginUpdate** e **EndUpdate**: **AddField** coloca uma coluna nas linhas, nas colunas ou nos filtros, com um agrupamento de datas opcional, e **AddDataField** adiciona uma medida com sua agregação.
- Crie uma **TsgcWinFormsPivotFieldList** e defina seu **PivotGrid** como a grade. O usuário arrasta os campos entre as quatro áreas e filtra os membros.

Exemplo

Enter ou um clique duplo em uma célula dispara **OnDrillDown**; **DrillDown** do motor devolve as linhas de origem da célula como uma tabela que uma grade sgcGrid ou um segundo pivot pode mostrar, veja Vinculação de dados no .NET. Um clique no botão +/- de um membro, ou um clique duplo no cabeçalho de uma linha, expande e recolhe o membro.

Veja também

- Instalação pelo NuGet
- Início rápido WPF
- Início rápido Avalonia
- Classes do .NET
- TsgcVCLPivotGrid
- Drill-down
- Demos do .NET

Início rápido WPF

O mesmo pivot no WPF, montado em código: um **DockPanel** com uma caixa de seleção de tema no topo, a lista de campos à direita e a grade no restante da janela. A coluna Date vai duas vezes para as colunas, agrupada por ano e por trimestre, e a área de valores contém duas medidas da mesma coluna, a soma e a média das vendas. Um clique em uma célula mostra o texto dela na barra de título.

Passos

- Crie um projeto WPF `net8.0-windows` e adicione o pacote **esegece.sgcPivotGrid.Wpf**.
- Crie um **TsgcWpfPivotGrid** e uma **TsgcWpfPivotFieldList**, adicione ambos à janela e defina **PivotGrid** da lista de campos como a grade.
- Preencha **Pivot** como no exemplo de WinForms. O motor, o layout e os eventos são os mesmos em todos os hosts.
- Leia o resultado entre **Lock** e **Unlock** do motor. **Result** é um instantâneo imutável, de modo que um recálculo nunca altera as células que você está lendo.

Exemplo

A grade WPF desenha suas próprias barras de rolagem, como a grade FireMonkey, e trabalha em pixels independentes de dispositivo: **HitTest** recebe DIPs. As opções de exibição são propriedades CLR simples, defina-as em código ou como atributos XAML.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcFMXPivotGrid

Início rápido Avalonia

O mesmo pivot no Avalonia, para Windows, Linux e macOS: as quantidades de quatro produtos por região. Uma caixa de seleção está vinculada a **AlternateRows** da grade, e o título acompanha a linha selecionada.

Passos

- Crie um projeto desktop Avalonia `net8.0` com **Avalonia.Desktop** e **Avalonia.Themes.Fluent** 11.3, e adicione o pacote **esegece.sgcPivotGrid.Avalonia**.
- Adicione o **FluentTheme** aos estilos do aplicativo: o popup de membros da lista de campos usa os templates Fluent das caixas de seleção e da caixa de pesquisa.
- Crie um **TsgcAvaloniaPivotGrid** e uma **TsgcAvaloniaPivotFieldList**, defina **PivotGrid** da lista de campos como a grade e preencha **Pivot** como nos outros hosts.

Exemplo

Cada opção de exibição da grade Avalonia é uma propriedade estilizada, portanto aceita vinculações, estilos e valores XAML, e **SelectedRow** e **SelectedCol** são propriedades diretas bidirecionais. No macOS a tecla Cmd funciona como Ctrl para Ctrl+Home e Ctrl+End, e uma roda horizontal rola as colunas.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Vinculação de dados no .NET
- Diferenças do .NET em relação ao Delphi
- TsgcFMXPivotGrid

Classes do .NET

Todas as classes ficam no namespace **esegece.sgcPivotGrid**. As grades e as listas de campos são hosts finos sobre o motor, o renderizador e o modelo de lista de campos independentes de framework, compartilhados pelos hosts, pela saída sem interface e pelas origens.

Classes de host

Host	Classe	Classe 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

A lista de seleção de membros de cada lista de campos é **TsgcWinFormsPivotMemberPopup** (um Form), **TsgcWpfPivotMemberPopup** (uma Window) ou **TsgcAvaloniaPivotMemberPopup** (um Popup). As três grades implementam **IsgcPivotEngineHost**, para que uma origem ou uma lista de campos encontre o motor delas, e **IsgcUITable**, para que um gráfico do sgcCharts ou outro pivot possa ler o resultado delas.

Membros da grade

As três grades compartilham os mesmos membros.

Kind	Membros
Motor e desenho	**Pivot** (o **TsgcPivotEngine**), **Renderer**, **FormatRules**.
Opções de exibição	**RowLayout**, **RowHeight**, **HeaderHeight**, **IndentWidth**, **MinColumnWidth**, **MaxColumnWidth**, **AutoColumnWidths**, **GridLines**, **AlternateRows**, **TotalsBold**, **ShowExpandButtons**, **Theme**, **AccentColor**, **FontSize**.
Posição	**TopRow**, **ScrollX**, **SelectedRow**, **SelectedCol**, **ScrollToCell**, **HitTest**.
Saída	**Recalculate**, **SaveToPNG** e **SaveToSVG** (um arquivo ou um stream, com largura e altura opcionais; 0 é a visão inteira), **SaveToCSV**, **SaveToJSON**, **SaveToHTML**. Edição Registered: **SaveLayoutToString**, **LoadLayoutFromString**, **SaveLayoutToFile**, **LoadLayoutFromFile**.
Eventos	**OnCellClick**, **OnCellDblClick**, **OnDrillDown**, **OnExpandChange**, **OnGetCellStyle**, **OnGetCellText**, **OnSelectionChange**, **OnColumnResized**.

As listas de campos têm **PivotGrid** (qualquer **IsgcPivotEngineHost**), **Engine**, **Theme**, **AccentColor**, **FontSize**, **ShowFilterArea**, **DeferUpdate** com **ApplyChanges**, e os eventos **OnBeforeFieldMove** e **OnFieldMoved**.

Tipos independentes de framework

Tipo	Descrição
TsgcPivotEngine	O armazenamento de colunas, o layout, o cálculo, o instantâneo, o drill-down e a exportação: **BeginLoad**, **DefineColumn**, **AppendRow**, **EndLoad**, **LoadFromTable**, **AddField**, **AddDataField**, **MoveField**, **RemoveField**, **ClearLayout**, **ExpandAll**, **CollapseAll**, **Lock**, **Unlock**, **Result**, **ResultTable**, **GetSourceRows**, **DrillDown**, **GetMemberLabels** e os eventos **OnCalculated**, **OnLayoutChange** e **OnFilterRow**. Veja TsgcPivotEngine.
TsgcPivotFields, TsgcPivotField	Um campo do layout: **FieldName**, **Caption**, **Area**, **AreaIndex**, **DateGrouping**, **Aggregation**, **DisplayFormat**, **SortBy**, **SortOrder**, **SortMeasure**, **Subtotals**, **Expanded**, **FilterValues**, **FilterExclude**, **LabelFilter**, **ValueFilter** e, na edição Registered, **TopN**, **TopNOthers**, **ShowAs**, **ShowAsAxis** e **ShowAsBaseItem**. Veja TsgcPivotField.
TsgcPivotCalculatedFields, TsgcPivotCalculatedField	Colunas numéricas calculadas por linha de origem a partir de uma expressão. Edição Registered. Veja TsgcPivotCalculatedField.
TsgcPivotResult, TsgcPivotItem	O instantâneo imutável de um recálculo: **ItemCount**, **Item**, **NodeLabel**, **NodePath**, **LevelCount**, **MeasureCount**, **DataColCount**, **CellText**, **CellValue** e **RawValue**. Um **TsgcPivotItem** é um struct com **Kind** (`pvkMember` , `pvkTotal` , `pvkGrandTotal`), **Level**, **Node**, **Expandable** e **Expanded**.
TsgcPivotResultTable, TsgcPivotSourceView	O resultado de um motor e as linhas de origem de um drill-down, ambos como **IsgcUITable**.
TsgcPivotSource_DataSet	Carrega uma `System.Data.DataTable` e a recarrega quando a tabela muda. Veja Vinculação de dados no .NET.
TsgcPivotSource_Table	Carrega qualquer **IsgcUITable** e a recarrega quando sua versão muda. Veja TsgcPivotSource_Table.
TsgcPivotGridRenderer, TsgcPivotRenderOptions, TsgcPivotRenderState	Desenha um instantâneo em um **TsgcUICanvas**, faz os testes de acerto e o grava como SVG ou, com esegece.sgcPivotGrid.Rendering, como PNG. Veja Renderização sem interface no .NET.
TsgcPivotFormatRules, TsgcPivotFormatRule	Formatação condicional: regras de valor e escalas de cor. Veja TsgcPivotFormatRule.
TsgcPivotFieldListModel	A lista de campos sem janela: as áreas, os chips, arrastar e soltar e o estado dos filtros, desenhados pelas três listas de campos.
sgcPivotGrid_IO	Exportação estática para CSV, JSON e HTML de um resultado, com os argumentos padrão do Delphi. O motor e as grades a encapsulam em **SaveToCSV**, **SaveToJSON** e **SaveToHTML**.

Enumerações

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`. Edição Registered.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- Vinculação de dados no .NET
- Renderização sem interface no .NET
- Arquitetura

Diferenças do .NET em relação ao Delphi

O port .NET mantém os nomes, os padrões, a aritmética e as regras de renderização do Delphi. Sua saída SVG e a exportação para CSV, JSON e HTML de dados comuns são idênticas byte a byte às do Delphi. As diferenças abaixo vêm dos frameworks dos hosts e da biblioteca de classes do .NET.

Sem streaming de componentes

Não há DFM no .NET, portanto não há editores de coleção em tempo de design, nem lista suspensa de FieldName, nem os verbos **Recalculate** e **Clear Layout**. Monte o layout em código com **AddField** e **AddDataField**, entre **BeginUpdate** e **EndUpdate**, para que o motor recalcule uma única vez. O motor Delphi adia o trabalho enquanto um formulário carrega; no .NET isso é função de **BeginUpdate** e **BeginLoad**. Para manter um layout entre sessões use o JSON de layout da edição Registered, veja Persistência do layout.

Propriedades indexadas, eventos e listas

- Uma propriedade indexada do Delphi é um método: `Result.Item(axis, i)`, `NodeLabel(axis, node)`, `SourceColumnName(i)` e, nos popups de membros, `Members(i)` e `MemberChecked(i)` com `SetMemberChecked(i, value)`. Uma função do Delphi continua sendo um método: `SourceRowCount()`, `MeasureCount()`, `DataColCount()`. As coleções mantêm seu indexador, como `Fields[0]`.
- Os eventos são propriedades delegate com os nomes do Delphi (**OnCalculated**, **OnDrillDown**...). Atribua um manipulador com `=` ou encadeie vários com `+=`. Um parâmetro `var` do Delphi é um parâmetro `ref` (`aHandled`, `aAccept`, `aAllow`), e uma lambda com um parâmetro `ref` declara o tipo de todos os parâmetros. **TNotifyEvent** é **TsgcUINotifyEvent**.
- Um parâmetro **TStrings** é uma `IList<string>`. **FilterValues** é um **TsgcPivotFilterValues**, uma lista ordenada de strings cujo **IndexOf** ignora maiúsculas e minúsculas, como TStringList. O filtro em si compara os rótulos dos membros diferenciando maiúsculas de minúsculas.
- **Assign** de um campo, um filtro, uma regra ou um campo calculado a partir de outra classe lança `InvalidCastException`, o EConvertError do Delphi.

Valores, datas e culturas

AppendRow recebe valores CLR: null e DBNull são células vazias, os tipos inteiros são `pvtInteger`, `float`, `double` e `decimal` são `pvtFloat`, `DateTime` é `pvtDateTime`, `bool` é `pvtBoolean`, e `string` e `char` são texto. As datas são armazenadas como datas OLE, o TDateTime do Delphi, de modo que o agrupamento de datas e os números de semana coincidem com o Delphi. O texto é analisado apenas com a cultura invariante, enquanto o Delphi também tenta a localidade do usuário, e os rótulos dos membros, **CellText** e as exportações formatam números e datas com configurações invariantes (ponto como separador decimal, datas como MM/dd/yyyy), nunca com a cultura da thread.

DataTable em vez de TDataSet

TsgcPivotSource_DataSet vincula uma `System.Data.DataTable`, o equivalente .NET de um TDataSet do Delphi. Uma DataTable está sempre aberta e não tem cursor, portanto não há registro atual, nem marcador, nem DisableControls. Os eventos `RowChanged`, `RowDeleted` e `TableCleared` ocupam o lugar do data link do Delphi; edições inacabadas e novas linhas desanexadas são ignoradas, assim como o Delphi ignora edições e inserções não confirmadas. Veja Vinculação de dados no .NET.

Sem notificação de liberação

Quando **PivotGrid** de uma origem ou de uma lista de campos é um `IComponent`, como a grade WinForms, a origem escuta o evento `Disposed` dela e então limpa **PivotGrid**. As grades WPF e Avalonia não são componentes: defina **PivotGrid** como null antes de descartar a grade. **Engine** nunca é monitorado; limpe-o antes que o motor desapareça. **TsgcPivotSource_DataSet** e **TsgcPivotSource_Table** são `IDisposable`; **Dispose** remove seus ouvintes da tabela.

Threads

O cálculo é síncrono, como no Delphi. Um recálculo roda na thread que o dispara e publica nela o novo instantâneo; as grades então se redesenham na thread de interface, por meio de uma mensagem de janela postada no WinForms, `Dispatcher.BeginInvoke` no WPF e `Dispatcher.UIThread.Post` no Avalonia. O código que lê **Result** de outra thread mantém **Lock** e **Unlock** do motor.

Barras de rolagem, unidades e propriedades por host

- **TsgcWinFormsPivotGrid** usa as barras de rolagem nativas do Windows e pixels, como a grade VCL. **TsgcWpfPivotGrid** e **TsgcAvaloniaPivotGrid** desenham suas próprias barras de rolagem, com arrasto do polegar e paginação na trilha, como a grade FireMonkey, e trabalham em pixels independentes de dispositivo.
- O **BorderStyle** do Delphi é **BorderThickness** (1 por padrão) e **BorderBrush** no WPF e no Avalonia. **TsgcWpfPivotGrid** declara seu próprio **FontSize** com `new`, e **TsgcAvaloniaPivotGrid** seu próprio **Theme** do tipo **TsgcUITheme**, que oculta o tema de controle do Avalonia. Por meio de uma referência do tipo `Control` são as propriedades do framework, que a grade não lê.
- As opções do WPF são propriedades CLR. As opções do Avalonia são propriedades estilizadas que aceitam vinculações e estilos, veja Vinculação de dados no .NET.
- O popup de membros da lista de campos é um Form no WinForms, uma Window sem borda pertencente à janela da lista de campos no WPF, e um Popup de fechamento por clique externo no Avalonia. No macOS a tecla Cmd funciona como Ctrl.

Veja também

- Visão geral do .NET
- Classes do .NET
- Vinculação de dados no .NET
- Início rápido WPF
- Início rápido Avalonia
- TsgcPivotEngine

Vinculação de dados no .NET

O motor mantém sua própria cópia tipada das linhas de origem, de modo que um pivot nunca lê a fonte de dados enquanto calcula ou desenha. Há quatro formas de preenchê-lo:

Forma	Use para
BeginLoad, **DefineColumn**, **AppendRow**, **EndLoad**	Linhas que seu código produz ou lê de qualquer leitor. Todo início rápido a usa.
TsgcPivotSource_DataSet	Uma `System.Data.DataTable`, recarregada quando suas linhas mudam.
TsgcPivotSource_Table	Qualquer **IsgcUITable**: as grades e o armazenamento de linhas do sgcGrid .NET, outro motor pivot ou seu resultado, recarregado quando a versão da tabela muda.
LoadFromTable	Uma cópia única de uma **IsgcUITable**, como as linhas de origem de um drill-down.

Uma origem preenche o motor de uma grade quando **PivotGrid** está definido, ou um motor sem interface quando **Engine** está definido; **Engine** prevalece quando ambos estão definidos. Cada carga substitui as linhas em um único bloco de carga, de modo que o motor recalcula uma vez e mantém seu layout.

DataTable

O exemplo imprime `4 source rows, 3 columns`: a coluna Notes não está em **Columns**, e a linha adicionada após a vinculação recarregou o motor. O tipo de uma coluna da DataTable fixa o tipo da coluna do pivot:

Tipo da coluna da DataTable	Tipo da coluna do pivot
`byte` , `short` , `int` , `long` e os demais tipos inteiros	`pvtInteger`
`float` , `double` , `decimal`	`pvtFloat`
`DateTime` , `DateTimeOffset` , `TimeSpan`	`pvtDateTime`
`bool`	`pvtBoolean`
Colunas binárias (`byte[]` e os demais arrays, exceto `char[]`)	Nunca carregadas.
Qualquer outro	`pvtText` , com a cultura invariante.

Com **LiveUpdates** uma alteração confirmada recarrega o motor: uma linha adicionada, alterada, revertida ou excluída, e uma tabela limpa. Uma linha em modo de edição carrega seus valores atuais, de modo que uma carga nunca confirma uma edição pendente, e uma nova linha é ignorada até ser adicionada à tabela. **Load** carrega a qualquer momento e **Refresh** recarrega quando **Active** é true.

IsgcUITable

PivotOfCell cria um segundo pivot sobre as linhas por trás de uma célula, para uma janela de detalhe. **Follow** cria um pivot de um pivot: um motor é uma **IsgcUITable** de seu resultado, com uma coluna por campo de linha e uma por coluna de dados, nomeada de acordo com os membros de coluna e a medida, por exemplo `Online | Sum of Sales`. Cada recálculo do primeiro motor altera a versão da tabela e recarrega o segundo. A mesma propriedade **Table** aceita uma grade sgcGrid .NET, cujas linhas um usuário edita, como faz a demo GridToPivotToChart.

Vinculando as opções de exibição

No Avalonia cada opção de exibição da grade (**RowLayout**, **RowHeight**, **HeaderHeight**, **IndentWidth**, **MinColumnWidth**, **MaxColumnWidth**, **AutoColumnWidths**, **GridLines**, **AlternateRows**, **TotalsBold**, **ShowExpandButtons**, **Theme**, **AccentColor**, **FontSize**, **BorderThickness**, **BorderBrush**) é uma propriedade estilizada, e **SelectedRow** e **SelectedCol** são propriedades diretas bidirecionais. Elas aceitam vinculações, estilos e valores XAML, veja Início rápido Avalonia. Uma alteração feita em código em **Renderer.Options** é copiada de volta para as propriedades. A lista de campos do Avalonia também tem **Theme**, **AccentColor**, **FontSize**, **ShowFilterArea** e **DeferUpdate** como propriedades estilizadas.
No WPF e no WinForms as opções são propriedades CLR: defina-as em código, ou como atributos em XAML no WPF.

Veja também

- Visão geral do .NET
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcPivotSource_DataSet
- TsgcPivotSource_Table
- Interoperabilidade com gráficos

Renderização sem interface no .NET

O motor e o renderizador não precisam de janela, de modo que um serviço, um aplicativo web ou um job de console no Windows, Linux ou macOS pode criar um pivot e gravá-lo como PNG, SVG, CSV, JSON ou HTML. Este exemplo carrega 10.000 linhas de vendas, mantém o canal online com um campo de filtro, imprime o resultado, lê as linhas de origem por trás de uma célula e grava os cinco arquivos.

Exemplo

A saída do console:

```
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
```

Observações

- Uma célula é endereçada por um item de linha e uma coluna de dados. Com várias medidas, as colunas de dados de um item de coluna ficam lado a lado: a coluna de dados é o item de coluna vezes **MeasureCount** mais a medida.
- **TsgcPivotRenderState** guarda a posição de rolagem, a seleção e as larguras das colunas; `sgcPivotGrid_Render.sgcPivotRenderState()` devolve o estado padrão. Uma largura e uma altura de 0, o padrão, desenham a visão inteira em vez de uma tela.
- **SaveToSVG** faz parte de esegece.sgcPivotGrid e não tem dependência nativa. **SaveToPNG** é um método de extensão de esegece.sgcPivotGrid.Rendering, que desenha com o SkiaSharp.
- As grades têm os mesmos métodos: `Grid.SaveToPNG(fileName)` grava a visão inteira da grade com suas opções e regras de formato.
- A exportação para CSV grava os valores brutos com precisão total; a exportação para HTML grava os textos das células formatados.

Veja também

- Visão geral do .NET
- Classes do .NET
- TsgcPivotGridRenderer
- Exportação
- Início rápido headless
- Demos do .NET

Demos do .NET

As demos ficam em `.net\demos\PivotGrid`, uma pasta por demo com seu projeto, seu `Program.cs` e um README.md. Cada demo tem uma chave `--test` que executa uma autoverificação fora da tela, grava linhas PASS e FAIL e termina com o número de verificações que falharam, como a chave `/test` das demos Delphi.

Demo	Host	Mostra
SalesPivot	WinForms	100.000 linhas de vendas geradas com uma lista de campos: Region e Country nas linhas, a coluna Date como Year, Quarter e Month nas colunas, o canal como filtro, ordenação por valor, uma escala de cor, expandir e recolher, e exportação para CSV, HTML e PNG.
SalesPivotWpf	WPF	O mesmo pivot no **TsgcWpfPivotGrid**.
SalesPivotAvalonia	Avalonia	O mesmo pivot no **TsgcAvaloniaPivotGrid**, para Windows, Linux e macOS.
GridToPivotToChart	WinForms	Uma grade sgcGrid .NET, editável na edição Registered, alimenta o pivot por meio de **TsgcPivotSource_Table**, e o resultado do pivot alimenta um gráfico sgcCharts .NET por meio de **IsgcUITable**.
MillionRows	WinForms	Um milhão de linhas geradas com três medidas (a soma, a contagem distinta de 100.000 clientes e o desvio padrão), com o tempo da carga e de cada recálculo.
HeadlessReport	Console	Lê um arquivo CSV, ou primeiro grava um orders.csv de exemplo com 5.000 pedidos gerados, e grava o pivot como PNG e SVG com **TsgcPivotGridRenderer**, e como HTML, CSV e JSON com o motor, sem janela. Roda no Windows, Linux e macOS.

Compilando as demos

Dentro da árvore do produto as demos são compiladas contra os projetos da biblioteca. O zip das demos usa por padrão os pacotes NuGet da edição Community. Nenhuma usa um recurso avançado, portanto toda demo compila e roda nas duas edições. Compile uma demo com os pacotes Registered definindo **SgcPivotGridEdition**:

```
dotnet build SalesPivot\SalesPivot.csproj -p:SgcPivotGridEdition=Registered
dotnet run --project HeadlessReport\HeadlessReport.csproj -- --test
```

GridToPivotToChart também restaura os pacotes sgcGrid e sgcCharts da mesma edição.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Início rápido WinForms
- Renderização sem interface no .NET
- Demos

Solução de problemas

A grade permanece vazia

Verifique se os dados chegaram: SourceRowCount do engine é 0 quando a origem não está Active ou não tem DataSet, Table nem PivotGrid. Verifique os nomes: um campo de linha, coluna ou filtro cujo FieldName não corresponde a nenhuma coluna de origem é ignorado, e um campo de dados com nome desconhecido permanece vazio. Os nomes são comparados ignorando maiúsculas e minúsculas, e SourceColumnName lista as colunas carregadas. Com AutoRecalculate False, chame Recalculate. Um filtro que não deixa nenhuma linha de origem também resulta em uma tabela dinâmica vazia.

A tabela dinâmica não acompanha o dataset

TsgcPivotSource_DataSet recarrega enquanto Active e LiveUpdates são True, quando o dataset informa uma alteração de dados no estado de navegação: um post, um append, um delete, um refresh, First ou Last. Next, Prior e MoveBy não recarregam, e uma edição só recarrega depois de ser enviada com post.

O carregamento é lento

AppendRow fora de um bloco BeginLoad e EndLoad recalcula após cada linha. Carregue dentro de um bloco, veja Desempenho.

EsgcPivotError: The pivot layout needs more than ... cells

O layout cria mais células do que MaxCells, normalmente porque um campo com muitos membros está nas linhas e outro nas colunas. Agrupe as datas, mova um campo para os filtros, use Top N ou aumente MaxCells.

EsgcPivotError: Calculated field ...

A expressão desse campo calculado tem um erro de sintaxe, ou os campos calculados se referem uns aos outros em círculo. A mensagem informa o nome do campo. Um nome de campo que ainda não foi carregado não é um erro, o campo calculado permanece vazio até que a coluna chegue. Veja Campos calculados.

EsgcPivotError: Invalid pivot layout

LoadLayoutFromString ou LoadLayoutFromFile recebeu um JSON malformado, ou uma raiz que não é um objeto. O layout permanece inalterado.

Undeclared identifier CalculatedFields, TopN, ShowAs or SaveLayoutToString

A edição Community deixa de fora campos calculados, mostrar valores como, Top N e persistência do layout, portanto esses membros não existem nela, junto com OthersCaption, TopNOthers, GetExpandState e SetExpandState. Um formulário que armazena essas propriedades não pode ser carregado por um build Community. Veja Edições.

Células vazias não mostram nada

Defina EmptyText do engine, por exemplo como um traço. As grades, o renderer, a exportação HTML e ResultTable o exibem, e OnGetCellText o recebe e ainda pode substituí-lo. SaveToCSV grava um campo vazio e SaveToJSON grava null, de modo que os dados exportados continuam numéricos.

Um total não é a soma das células

Os totais são calculados a partir das linhas de origem: uma média é a média das linhas, e uma contagem distinta conta cada valor uma única vez. Após um filtro de valor ou Top N, os totais incluem apenas os membros que permanecem. Um campo calculado é avaliado por linha e depois agregado. Veja Agregações e Totais.

As datas não agrupam, ou os números são ordenados como texto

DateGrouping só funciona em uma coluna de data, e uma ordenação por rótulo compara números pelo valor apenas em uma coluna numérica. Uma coluna cujo primeiro valor foi uma string é uma coluna de texto. Passe valores TDateTime e numéricos a AppendRow, ou fixe o tipo com DefineColumn(aName, aType) antes da primeira linha.

Uma visão de drill-down está vazia, ou uma violação de acesso ao encerrar

Um TsgcPivotSourceView fica sem linhas depois que os dados de origem mudam, e ele lê o armazenamento do seu engine, portanto libere-o antes do engine. Veja Drill-down.

Valores errados ou uma exceção quando outra thread lê Result

Leia Result somente entre Lock e Unlock do engine. Um recálculo publica um novo snapshot e libera o anterior.

C++Builder: no matching constructor for TsgcPivotEngine or TsgcPivotGridRenderer

Os cabeçalhos gerados não mantêm os argumentos padrão desses construtores. Passe NULL explicitamente: `new TsgcPivotEngine(NULL)` e `new TsgcPivotGridRenderer(NULL, NULL)`.

TsgcFMXPivotGrid is missing or empty

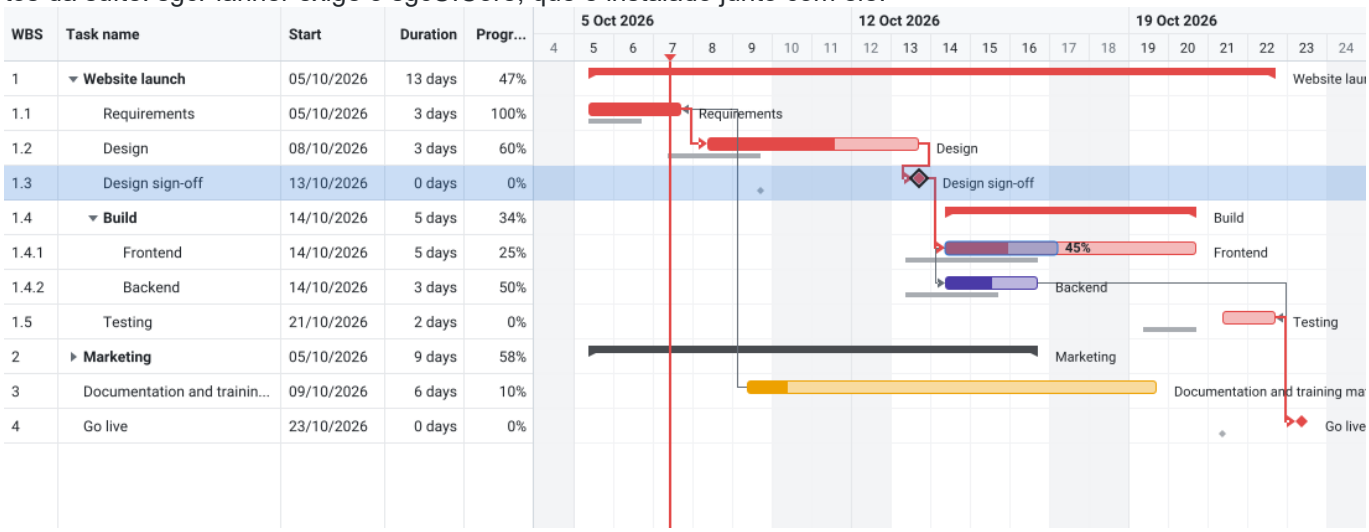
TsgcFMXPivotGrid e TsgcFMXPivotFieldList existem a partir do Delphi XE6, em que o sgcVer.inc define SGC_PIVOTGRID_FMX. Com um compilador mais antigo, a unit sgcPivotGrid_FMX compila sem eles. Em tempo de design, estão na página SGC PivotGrid da paleta enquanto um formulário FireMonkey está aberto. Veja Plataformas.

A lista de campos não mostra campos

Defina PivotGrid da lista de campos, ou Engine para um engine sem grade. A lista mostra as colunas de origem do engine, portanto é preenchida quando os dados são carregados.

Introdução

sgcPlanner adiciona agendamento e planejamento de projetos a aplicações Delphi e C++Builder, para VCL e FireMonkey. Um objeto de dados, **TsgcPlannerData**, guarda os itens, os recursos, os status, as dependências e o calendário de trabalho, e três tipos de visão o mostram ao mesmo tempo: **TsgcVCLScheduler** e **TsgcFMXScheduler** (dia, semana útil, semana, mês, agenda e uma Timeline de recursos), **TsgcVCLGantt** e **TsgcFMXGantt** (uma árvore de tarefas com barras, dependências, agendamento automático, caminho crítico e linhas de base) e **TsgcVCLKanban** e **TsgcFMXKanban** (um quadro com colunas de status, raias e limites de WIP). Arraste um cartão para outra coluna e o Gantt e o Scheduler mostram o novo status na hora.
Os dados vêm do seu código, de um **TDataSet** por meio de **TsgcPlannerSource_DataSet**, de um arquivo JSON, de um arquivo iCalendar (.ics) por meio de **TsgcPlannerICalendar** ou de um arquivo XML do Microsoft Project por meio de **TsgcPlannerMSProject**. **TsgcPlannerRenderer** desenha qualquer visão em PNG e SVG sem janela, em um servidor ou no Linux64.
sgcPlanner faz parte da suíte visual da eSeGeCe. Ele é desenhado com **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro, de modo que um planner tem a mesma aparência no VCL, no FireMonkey e na saída sem janela, e combina com os outros produtos da suíte. sgcPlanner exige o sgcUICore, que é instalado junto com ele.



Edições

sgcPlanner é distribuído nas edições Registered, Trial e Community. A edição Community mantém as três visões, o arrastar e soltar, os resumos, os marcos, o progresso, a origem DataSet, o JSON e a saída sem janela, e deixa de fora a recorrência, o agendamento automático com setas de dependência, o caminho crítico, as linhas de base, a Timeline de recursos, as raias, os limites de WIP, o iCalendar e o XML do MS Project. Veja Edições.

Por onde começar

- Arquitetura: o objeto de dados, as notificações de alteração, as visões e as plataformas.
- Início Rápido VCL, Início Rápido FMX e Início Rápido C++Builder: três visões sincronizadas em poucos minutos.
- TsgcPlannerData: os itens, os recursos, os status, as dependências e o calendário de trabalho.
- As visões: TsgcVCLScheduler, TsgcVCLGantt, TsgcVCLKanban e seus equivalentes FireMonkey.
- Os componentes não visuais: TsgcPlannerSource_DataSet, TsgcPlannerICalendar, TsgcPlannerMSProject e TsgcPlannerRenderer.
- Instalação e Plataformas: os pacotes, as pastas e as IDEs e plataformas suportadas.
- Demos: as oito demos Delphi e o que cada uma mostra.
- Os recursos: Dados Compartilhados, Visões Sincronizadas, Calendário de Trabalho, Recorrência, Escala de Tempo e Zoom, Arrastar e Soltar, Dependências, Agendamento Automático, Caminho Crítico, Linhas de Base, Raias e Limites de WIP, Vínculo com DataSet, Persistência JSON, iCalendar, XML do MS Project, Renderização sem Janela, Temas e Recursos por Edição.
- Solução de Problemas: os problemas comuns e suas causas.

Arquitetura

sgcPlanner mantém os dados separados das visões, e as visões separadas do framework. O objeto de dados, os mecanismos e os modelos de visão usam apenas a RTL e o sgcUICore. Os controles VCL e FMX são hosts finos em torno de um modelo de visão, e o renderer sem janela usa os mesmos modelos de visão, de modo que um PNG coincide pixel a pixel com o que um formulário mostra.

Fluxo de dados

1. **Origens.** O seu código preenche o TsgcPlannerData diretamente, ou TsgcPlannerSource_DataSet o carrega a partir de TDataSets e grava as edições de volta. LoadFromJSON lê um projeto salvo, TsgcPlannerICalendar importa e exporta arquivos .ics e TsgcPlannerMSProject importa e exporta XML do Microsoft Project.
2. **Dados.** TsgcPlannerData é um TComponent com cinco partes: Items, Resources, Statuses, Dependencies e Calendar. Ele mantém um índice por Id, a estrutura de pais e filhos, a seleção e um número de Version que cresce a cada alteração.
3. **Mecanismos.** O calendário de trabalho conta minutos de trabalho, o mecanismo de recorrência expande regras RRULE em ocorrências e o mecanismo de agendamento coloca as tarefas após as suas predecessoras, calcula a folga e marca o caminho crítico. Os dois últimos fazem parte da edição Registered.
4. **Modelos de visão.** Cada visão se registra como listener dos dados. Em uma alteração ela reconstrói somente o que precisa: uma mudança de seleção redesenha um overlay, uma mudança de item refaz o layout no próximo quadro, uma mudança de estrutura reconstrói as linhas, as colunas e as raias.
5. **Saída.** Os modelos de visão desenham em um TsgcUICanvas do sgcUICore. Os seis controles os mostram em um formulário com rolagem, mouse, teclado e toque, e o TsgcPlannerRenderer os desenha em PNG e SVG.

Classes

Classe	Unit	Função
TsgcPlannerData	sgcPlanner_Data	Os dados compartilhados: itens, recursos, status, dependências, calendário, seleção, notificações e JSON.
TsgcPlannerItem	sgcPlanner_Data	Uma tarefa, cartão ou evento: datas, status, progresso, pai, recursos, tags, recorrência, restrição e linha de base.
TsgcPlannerCalendar	sgcPlanner_Calendar	Os dias úteis, o horário de trabalho, os feriados e a aritmética dos minutos de trabalho.
TsgcPlannerTimeScale	sgcPlanner_TimeScale	O eixo de tempo do Gantt e da Timeline: zoom, largura das marcas, marcas, faixas de tempo não útil e ajuste.
TsgcVCLScheduler, TsgcFMXScheduler	sgcPlanner_Scheduler_VCL, sgcPlanner_Scheduler_FMX	As visões de calendário: dia, semana útil, semana, mês, agenda e a Timeline de recursos.
TsgcVCLGantt, TsgcFMXGantt	sgcPlanner_Gantt_VCL, sgcPlanner_Gantt_FMX	A árvore de tarefas e o gráfico de barras com dependências, caminho crítico e linhas de base.
TsgcVCLKanban, TsgcFMXKanban	sgcPlanner_Kanban_VCL, sgcPlanner_Kanban_FMX	O quadro: colunas de status, raias, limites de WIP e cartões.

TsgcPlannerSource_DataSet	sgcPlanner_Source_DataSet	Carrega os dados de TDataSets, acompanha as alterações deles e grava as edições de volta.
TsgcPlannerICalendar	sgcPlanner_ICal	Importação e exportação de VEVENT da RFC 5545. Edição Registered.
TsgcPlannerMSProject	sgcPlanner_MSProject	Importação e exportação de XML do Microsoft Project (MSPDI). Edição Registered.
TsgcPlannerRenderer	sgcPlanner_Renderer	Desenha uma visão Scheduler, Gantt ou Kanban sem janela, em um canvas, um PNG ou um SVG.
TsgcPlannerScheduleEngine	sgcPlanner_AutoSchedule	Passagem para a frente e para trás, folga, caminho crítico, linhas de base e detecção de ciclos. Edição Registered.

Notificações de alteração

- Todo setter dos dados notifica. Envolva um lote de alterações em BeginUpdate e EndUpdate: os listeners e o OnChange são acionados uma vez, após o EndUpdate mais externo, com os tipos de alteração (pckItems, pckStructure, pckResources, pckStatuses, pckDependencies, pckCalendar, pckSelection, pckSchedule), o Id do item quando exatamente um item mudou e a Version.
- As visões são listeners e apenas leem os dados. O seu handler de OnChange pode alterar os dados novamente: as alterações aninhadas chegam como mais uma rodada após a atual, até 8 rodadas, e então EsgcPlannerError é gerado.
- Uma edição feita com o mouse aciona OnItemChanging dos dados uma vez por drop, com os campos prestes a mudar, de modo que a aplicação pode recusá-la. A edição é então confirmada em uma única atualização, de modo que um drop dá uma notificação a cada visão vinculada.
- Duas ou mais visões sobre os mesmos dados permanecem em sintonia: com SyncSelection a seleção, que é SelectedId dos dados, também é compartilhada.

Tempo

O modelo guarda os seus horários como TsgcPlannerTime, uma contagem Int64 de minutos, de modo que a aritmética é exata e os segundos são descartados. As propriedades publicadas são TDateTime, e cada uma tem um equivalente TsgcPlannerTime: Start e StartTime, Finish e FinishTime, ProjectStart e ProjectStartTime. sgcPlannerTimeOf e sgcPlannerDateTimeOf de sgcPlanner_DateUtils convertem entre elas. O Finish de um item é exclusivo: um item de dia inteiro em 5 de outubro termina à meia-noite de 6 de outubro. As visões leem o horário atual da variável de função sgcPlannerNow, que um teste pode substituir para congelar o relógio.

Plataformas

Destino	Suporte
Delphi VCL	Delphi 7 a Delphi 13, Win32 e Win64.
Delphi FireMonkey	Delphi XE6 a Delphi 13 (SGC_PLANNER_FMX), toda plataforma que o FireMonkey suporta.
Linux64	Delphi 10.2 e posteriores para as units neutras em relação ao framework (SGC_PLANNER_LINUX): os dados, os mecanismos, JSON, iCalendar, XML do MS Project e o renderer sem janela.
C++Builder	C++Builder 2007 a C++Builder 13, com os mesmos componentes. Veja Início Rápido C++Builder.

Delphi 7 e Delphi 2007 têm strings Ansi, de modo que os componentes iCalendar e XML do MS Project convertem o texto UTF-8 dos arquivos para a página de código do sistema, e os caracteres fora dessa página de código são perdidos.

Desempenho

Estas são medições de builds release Win64 durante o desenvolvimento, em 1920x1080 e escala 1. O Win32 é cerca de 1,5 a 1,7 vez mais lento.

Cenário	Win64
Agendar automaticamente 10.000 tarefas (1.000 resumos, 3 níveis) com 15.000 vínculos e 30 feriados, caminho crítico incluído	11 ms
Quadro do Gantt ao rolar 10.000 tarefas com 15.000 vínculos	11 ms
Layout do Kanban de 5.000 cartões em 8 colunas e 6 raias	5 ms
Visão de semana do Scheduler com 2.000 itens com horário, layout	1 ms
Carregar 10.000 linhas com TsgcPlannerSource_DataSet	26 ms
Salvar e carregar JSON de 10.000 itens com 15.000 vínculos	48 ms e 56 ms
Importar 5.000 VEVENTs com fusos horários de um arquivo .ics	84 ms
Importar 10.000 tarefas com 15.000 vínculos do XML do MS Project	264 ms

Veja Também

- Edições
- TsgcPlannerData
- TsgcPlannerRenderer
- Plataformas
- Dados Compartilhados
- Visões Sincronizadas

Edições

sgcPlanner é compilado a partir do mesmo código-fonte para todas as edições.

Edição	Inclui
Registered	Tudo: as três visões para VCL e FMX, o objeto de dados, o calendário de trabalho, o arrastar e soltar, os resumos, os marcos, o progresso, a origem DataSet, o JSON, a saída PNG e SVG sem janela e os recursos avançados listados abaixo. A página da paleta SGC Planner mostra 11 componentes.
Trial	Os mesmos recursos da Registered. Um build compilado com SGC_TRIAL mostra uma mensagem de que é uma versão de avaliação quando um componente é criado depois que a janela de lançamento indicada no build passou. Nada mais muda.
Community	Tudo exceto os recursos avançados: as visões do Scheduler de dia, semana útil, semana, mês e agenda com agrupamento por recurso, o Gantt com a sua árvore de tarefas, resumos, marcos e progresso, o quadro Kanban com colunas de status, o arrastar e soltar em todas as visões, a origem DataSet, o JSON e o renderer. A paleta mostra 9 componentes, sem TsgcPlannerICalendar e TsgcPlannerMSProject, e não há aviso de avaliação.

Os recursos avançados

Os recursos avançados são compilados sob o define SGC_PLANNER_ADVANCED, que o sgcVer.inc desativa na edição Community. Seis units não fazem parte da distribuição Community: sgcPlanner_Recurrence, sgcPlanner_AutoSchedule, sgcPlanner_Scheduler_Timeline, sgcPlanner_XML, sgcPlanner_ICal e sgcPlanner_MSProject. As propriedades das visões são publicadas em todas as edições, de modo que na Community são mantidas mas não têm efeito:

Recurso	Na edição Community
Recorrência	GetOccurrences retorna um item recorrente uma vez, no seu próprio Start. RecurrenceRule e RecurrenceExceptions continuam armazenados e salvos. OnRecurrenceEdit e OnRecurrenceDelete do Scheduler nunca são acionados.
Dependências e agendamento automático	Os vínculos são armazenados e salvos, mas o Gantt não desenha setas e não tem arrastar para vincular. ShowDependencies e AutoSchedule não têm efeito, e Schedule e SetBaseline do Gantt geram EsgcPlannerError.
Caminho crítico e linhas de base	ShowCriticalPath e ShowBaseline não têm efeito. BaselineStart e BaselineFinish são apenas armazenados.
Timeline de recursos	A visão psvTimeline do Scheduler desenha um aviso centralizado.
Raias e limites de WIP	SwimlaneBy age como pslNone. WIPLimit não mostra contador e nunca bloqueia um cartão.
iCalendar e XML do MS Project	TsgcPlannerICalendar e TsgcPlannerMSProject não existem.

A edição é uma escolha em tempo de compilação no sgcVer.inc. SGC_COMMUNITY também desativa SGC_TRIAL. O código que usa TsgcPlannerICalendar, TsgcPlannerMSProject ou outra classe das seis units, e que também precisa compilar na Community, o coloca dentro de {$IFDEF SGC_PLANNER_ADVANCED}. Um formulário que contém um desses dois componentes não carrega em um build Community.

Veja Também

- Arquitetura
- Introdução
- Recursos por Edição
- Instalação

Instalação

sgcPlanner faz parte da distribuição sgcUI. Um instalador cobre todos os produtos sgcUI e instala apenas os produtos que a sua licença inclui.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder em que instalar.
3. Na página de componentes, selecione **sgcPlanner** e os outros produtos que quiser. Os produtos que a sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, compila os pacotes na sua máquina para cada IDE selecionada na ordem de dependência e instala o pacote de tempo de design. Os componentes aparecem na página da paleta **SGC Planner**.

O que vem com o sgcPlanner

sgcUICore, a base compartilhada dos produtos visuais, é instalado automaticamente com o sgcPlanner. Ele não tem caixa de seleção própria.

Pastas instaladas

Cada produto instala em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para a Trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de cada produto instalado, sgcPlanner e sgcUICore incluídos. Este é o único caminho de biblioteca do sgcUI.
`Packages`	Os pacotes de cada produto instalado, um conjunto por produto e versão de IDE.
`Lib<V>`	As units compiladas para cada versão de IDE e plataforma, por exemplo `LibD13`.
`Demos\Planner`	As demos do sgcPlanner. Veja Demos.

O instalador adiciona `sgcUI\Source` ao caminho de biblioteca e ao caminho de navegação de cada IDE e plataforma selecionada, junto com a pasta das units compiladas.

Trial e Community

- **Trial**: o instalador Trial do sgcUI instala o sgcPlanner em modo de avaliação, com todos os recursos.
- **Community**: o instalador Community gratuito do sgcUI inclui o sgcPlanner sem os recursos avançados (recorrência, agendamento automático, setas de dependência, caminho crítico, linhas de base, Timeline de recursos, raias, limites de WIP, iCalendar e XML do MS Project). Veja Edições.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão de IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo da versão do Delphi, por exemplo `sgcPlannerD13.dpk`.

Pacote	Conteúdo

`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcPlanner<V>`	O pacote runtime do sgcPlanner: os dados, os mecanismos, as visões, os controles VCL e, a partir do Delphi XE6, os controles FireMonkey. Ele exige rtl, vcl, dbrtl, fmx (XE6 e posteriores) e sgcUICore.
`dclsgcPlanner<V>`	O pacote de tempo de design que registra os componentes na página da paleta SGC Planner.

Compile-os na ordem de dependência: `sgcUICore`, depois `sgcPlanner`, depois `dclsgcPlanner`. Em seguida instale `dclsgcPlanner` e adicione `sgcUI\Source` ao caminho de biblioteca da sua versão do Delphi (Ferramentas > Opções > Linguagem > Opções do Delphi > Biblioteca > Caminho da biblioteca). Para o C++Builder, abra o projeto `.cbproj` da sua versão, por exemplo `sgcPlannerC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

Veja Também

- Plataformas
- Edições
- Início Rápido VCL

Plataformas

sgcPlanner é uma única árvore de código-fonte para toda IDE e plataforma. Os dados, os mecanismos e os modelos de visão usam apenas a RTL e o sgcUICore, de modo que são o mesmo código em todo destino; apenas os seis controles estão ligados a um framework.

Destino	Suporte
Delphi VCL	Delphi 7 a Delphi 13, Win32 e Win64: TsgcVCLScheduler, TsgcVCLGantt e TsgcVCLKanban.
Delphi FireMonkey	Delphi XE6 a Delphi 13 para Windows, macOS, iOS e Android: TsgcFMXScheduler, TsgcFMXGantt e TsgcFMXKanban. Compilados sob SGC_PLANNER_FMX.
Linux64	Delphi 10.2 e posteriores, compilados sob SGC_PLANNER_LINUX: os dados, o calendário de trabalho, os mecanismos, JSON, iCalendar, XML do MS Project e TsgcPlannerRenderer, de modo que um servidor pode produzir arquivos PNG e SVG de um plano.
C++Builder	C++Builder 2007 a C++Builder 13 com os mesmos componentes e os headers .hpp gerados. Veja Início Rápido C++Builder.
.NET	Um port em C# com a mesma API para WinForms, WPF e Avalonia, o Scheduler, o Gantt e o quadro Kanban em cada host e a saída PNG e SVG sem janela, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão geral do .NET.

Units por framework

Units	Framework
sgcPlanner_Const, sgcPlanner_DateUtils, sgcPlanner_Calendar, sgcPlanner_Data, sgcPlanner_TimeScale, sgcPlanner_Recurrence, sgcPlanner_AutoSchedule, sgcPlanner_JSON, sgcPlanner_XML, sgcPlanner_ICal, sgcPlanner_MSProject	Nenhum. O modelo, os mecanismos e os formatos de arquivo.
sgcPlanner_View, sgcPlanner_Scheduler, sgcPlanner_Scheduler_Render, sgcPlanner_Scheduler_Timeline, sgcPlanner_Gantt, sgcPlanner_Gantt_Render, sgcPlanner_Kanban, sgcPlanner_Kanban_Render, sgcPlanner_Renderer	Nenhum. Os modelos de visão e os seus desenhadores, que desenham em um TsgcUICanvas do sgcUICore.
sgcPlanner_Source_DataSet	Somente a unit DB (dbrtl), nenhuma unit VCL ou FireMonkey.
sgcPlanner_VCL, sgcPlanner_Scheduler_VCL, sgcPlanner_Gantt_VCL, sgcPlanner_Kanban_VCL	VCL, somente Windows.
sgcPlanner_FMX, sgcPlanner_Scheduler_FMX, sgcPlanner_Gantt_FMX, sgcPlanner_Kanban_FMX	FireMonkey, Delphi XE6 e posteriores.

Pacotes

O sgcPlanner traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote ter-

mina com o sufixo do IDE, por exemplo sgcPlannerD13.dpk para o Delphi 13 e sgcPlannerC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcPlanner	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey: os dados, o calendário de trabalho, os motores, os modelos de visão e seus painters, JSON, iCalendar, XML do MS Project, TsgcPlannerRenderer e TsgcPlannerSource_DataSet.	rtl, dbrtl, sgcUICore
sgcPlannerVCL	Runtime, VCL	sgcPlanner_VCL, sgcPlanner_Scheduler_VCL, sgcPlanner_Gantt_VCL e sgcPlanner_Kanban_VCL (TsgcVCLScheduler, TsgcVCLGantt, TsgcVCLKanban)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcPlanner
sgcPlannerFMX	Runtime, FireMonkey, a partir do XE6	sgcPlanner_FMX, sgcPlanner_Scheduler_FMX, sgcPlanner_Gantt_FMX e sgcPlanner_Kanban_FMX (TsgcFMXScheduler, TsgcFMXGantt, TsgcFMXKanban)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcPlanner
dclsgcPlanner	Tempo de projeto, VCL	Registra TsgcPlannerData, TsgcVCLScheduler, TsgcVCLGantt, TsgcVCLKanban, TsgcPlannerRenderer, TsgcPlannerSource_DataSet e, fora da edição Community, TsgcPlannerICalendar e TsgcPlannerMSProject na página **SGC Planner** da paleta.	designide, rtl, vcl, sgcPlanner, sgcPlannerVCL
dclsgcPlannerFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXScheduler, TsgcFMXGantt e TsgcFMXKanban na página **SGC Planner** da paleta.	designide, rtl, fmx, sgcPlanner, sgcPlannerFMX, dclsgcPlanner
sgcPlannerWS, dclsgcPlannerWS	Runtime e tempo de projeto, ponte do sgcWebSockets	TsgcPlannerGoogleSync e, fora da edição Community, TsgcPlannerICalSubscription, independentes de framework, na página **SGC Planner** da paleta.	rtl, sgcPlanner, sgcWebSockets

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcPlannerFMX exige o dclsgcPlanner, portanto instale primeiro o dclsgcPlanner.

Versões do IDE e plataformas de destino

Delphi	sgcPlanner	sgcPlannerVCL	sgcPlannerFMX	dclsgcPlanner, dclsgcPlannerFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE. O sgcPlannerWS é compilado para as mesmas plataformas do sgcPlanner, e o dclsgcPlannerWS para as mesmas plataformas do dclsgcPlanner.

C++Builder	sgcPlanner, sgcPlannerVCL	sgcPlannerFMX	dclsgcPlanner, dclsgcPlannerFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcPlanner)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcPlanner)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcPlannerC13.cbproj, sgcPlannerVCLC13.cbproj, sgcPlannerFMXC13.cbproj, dclsgcPlannerC13.cbproj e dclsgcPlannerFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. O sgcPlannerWS e o dclsgcPlannerWS são compilados para as mesmas versões do C++Builder e plataformas que o sgcPlanner e o dclsgcPlanner, exceto o Win64x, porque precisam de uma compilação Win64x do pacote sgcWebSockets. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXScheduler, TsgcFMXGantt e TsgcFMXKanban) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcPlannerVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcPlanner_Scheduler_FMX, sgcPlanner_Gantt_FMX ou sgcPlanner_Kanban_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcPlanner é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa os dados, o calendário de trabalho, os motores, JSON, iCalendar, XML do MS Project e TsgcPlannerRenderer, por exemplo para produzir arquivos PNG e

SVG de um plano. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Notas por plataforma

- **Alto DPI.** Tamanhos como RowHeight, TreeWidth e SlotHeight são em DIPs. Os controles VCL os escalam com o PPI do monitor, os controles FireMonkey com a escala do formulário e o renderer com a sua propriedade Scale.
- **Toque.** Os controles tratam a entrada por toque como entrada de mouse, de modo que um dedo arrasta um cartão, uma barra ou um evento. Os controles VCL mantêm os seus próprios gestos de pan, de modo que uma caixa de rolagem pai não assume um arrasto.
- **Locale.** Os nomes de dias e meses, os formatos de data e hora e o separador decimal vêm de sgcPlannerLocale, que é inicializado a partir da máquina. Um servidor Linux ou uma máquina com outro idioma mostra, portanto, outros rótulos. Defina as propriedades de sgcPlannerLocale na inicialização para ter a mesma saída em todo lugar.
- **Delphi 7 e Delphi 2007.** Essas versões têm strings Ansi, de modo que os leitores de JSON, iCalendar e XML do MS Project convertem o texto UTF-8 para a página de código do sistema, e os caracteres fora dela são perdidos.
- **Edições.** SGC_PLANNER_ADVANCED, que o sgcVer.inc desativa na edição Community, decide os recursos avançados em toda plataforma. Veja Recursos por Edição.

Veja Também

- Arquitetura
- Instalação
- Renderização sem Janela

Início Rápido VCL

Instale os pacotes do sgcPlanner para a sua versão do Delphi. Os componentes estão na página da paleta **SGC Planner**.

Em tempo de design

1. Solte um **TsgcPlannerData** em um formulário. Adicione status à sua coleção Statuses (por exemplo todo, doing e done), pessoas a Resources e tarefas a Items: um Id, um Title, um Start, um Finish e um Status para cada uma.
2. Solte um **TsgcVCLGantt**, defina Align como alTop e defina a sua propriedade Data como o objeto de dados.
3. Solte um **TsgcVCLKanban**, defina Align como alRight e defina Data. O quadro mostra uma coluna por status.
4. Solte um **TsgcVCLScheduler**, defina Align como alClient, defina Data e escolha uma View, por exemplo psvWorkWeek.
5. Execute a aplicação. Arraste um cartão para outra coluna, uma barra no Gantt ou um evento no Scheduler: as três visões mostram a alteração na hora, porque compartilham os dados.

No código

Preencha os dados entre BeginUpdate e EndUpdate, para que cada visão receba uma notificação. O Finish de um item é 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;
```

Veja Também

- TsgcPlannerData
- TsgcVCLGantt
- TsgcVCLKanban
- TsgcVCLScheduler
- Demos
- Instalação

Início Rápido FMX

Os componentes FireMonkey exigem o Delphi XE6 ou posterior. Eles estão na página da paleta **SGC Planner** de um formulário FireMonkey, junto com os componentes não visuais.

Em tempo de design

1. Solte um **TsgcPlannerData** e preencha-o como no Início Rápido VCL.
2. Solte um **TsgcFMXGantt**, um **TsgcFMXKanban** e um **TsgcFMXScheduler**, defina o Align deles e defina a propriedade Data de cada um como o objeto de dados.

Os controles FMX desenham as suas próprias barras de rolagem, rolam com a roda do mouse e tratam a entrada por toque como entrada de mouse, de modo que um dedo arrasta um cartão ou uma barra. As suas propriedades, métodos e eventos são os mesmos da VCL.

No 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;
```

Veja Também

- TsgcFMXScheduler
- TsgcFMXGantt
- TsgcFMXKanban
- Demos
- Plataformas

Início Rápido C++Builder

Instale os pacotes do sgcPlanner para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto e use os componentes da página da paleta **SGC Planner** exatamente como no Delphi. O instalador adiciona os dois caminhos à IDE para você. Os headers .hpp e os arquivos .lib gerados do sgcPlanner e do sgcUICore estão nessa única pasta, de modo que o sgcUICore não precisa de caminho próprio.

O que saber em C++

- Os construtores não têm argumentos padrão. Crie um componente não visual sem dono com NULL: `new TsgcPlannerRenderer(NULL)`.
- Os exemplos desta ajuda passam explicitamente todos os argumentos de um método, por exemplo `AddStatus(Id, Caption, 0)` e `AddDependency(FromId, ToId, pdkFS, 0)`, de modo que compilam com qualquer versão dos headers gerados.
- Os eventos são closures, de modo que os seus handlers são funções membro __fastcall de uma classe. Um parâmetro var como aAccept é uma referência bool, e um parâmetro const string é um `const String`.
- As enumerações não têm escopo: escreva `psvWorkWeek`, `prvGantt` ou `pwmBlock`.

No código

Veja Também

- TsgcPlannerData
- Início Rápido VCL
- TsgcPlannerRenderer
- Demos
- Plataformas

Demos

As demos estão em `sgcUI\Demos\Planner\Delphi`, uma pasta por demo com um `readme.html` que traz os detalhes. A maioria das demos monta os seus dados com um pequeno gerador determinístico em torno da semana de 5 de outubro de 2026.

Demo	Componentes	O que mostra
ProjectPlanner	TsgcVCLGantt, TsgcVCLKanban, TsgcVCLScheduler	O relançamento de um site em três visões vinculadas a um TsgcPlannerData. Arraste um cartão de To do para Done e a coluna de status do Gantt, a cor da barra e a cor do evento no Scheduler acompanham. Cores de status com BarColorMode pbcStatus e uma seleção compartilhada por SelectedId.
TeamScheduler	TsgcVCLScheduler	Três semanas de reuniões para cinco pessoas nas seis visões (dia, semana útil, semana, mês, agenda e a Timeline de recursos). Um stand-up diário e uma sincronização semanal a partir de uma RecurrenceRule, arrastar para criar com OnItemCreate e agrupamento com Options.GroupByResource.
ProjectGantt	TsgcVCLGantt, TsgcPlannerMSProject	Um plano de release de software com três fases EAP construídas a partir de ParentId, arrastar para vincular com OnLinkCreating, Schedule e AutoSchedule no calendário de trabalho, o caminho crítico, uma linha de base que mostra o quanto uma barra movida escorregou, e abrir e salvar XML do Microsoft Project. Edição Registered.
TaskBoard	TsgcVCLKanban	Um backlog de 26 cartões com larguras de coluna e limites de WIP em Columns, os modos de WIP de aviso e de bloqueio, raias por recurso, pai ou tag, navegação pelo teclado e uma coluna Done recolhível.
DataSetPlanner	TsgcVCLGantt, TsgcVCLKanban, TsgcPlannerSource_DataSet	Três tabelas TClientDataSet (tarefas, pessoas e vínculos) com os seus próprios nomes de campo, mapeadas com ItemFields, ResourceFields e DependencyFields. Um registro postado no TDBGrid atualiza o seu item, um cartão ou uma barra movida em uma visão é gravado na tabela, e as gravações esperam enquanto o grid edita um registro. MidasLib está vinculada, de modo que nenhum midas.dll é necessário.
ICalendarSync	TsgcVCLScheduler, TsgcPlannerICalendar	Exportar para .ics com a RRULE de uma série, itens de dia inteiro como valores DATE e horários UTC ou flutuantes com ExportUTC. Importar do Google Agenda, do Outlook ou de qualquer arquivo RFC 5545 com os fusos horários convertidos para o horário local, substituir ou mesclar com ImportMode e os avisos de OnImportWarning. Edição Registered.
GoogleCalendarSync	TsgcVCLScheduler, TsgcPlannerGoogleSync	Uma visão semanal em sincronização bidirecional com um calendário do Google por meio de TsgcHTTPGoogleCloud_Calendar_Client do sgcWebSockets: o id e o segredo do cliente OAuth2 digitados em tempo de execução e nunca armazenados, Direction, uma escolha para OnConflict, Sincronizar agora com SyncAsync, sincronização automática com AutoSyncInterval e um log de OnSyncDone, OnConflict e OnError. Edição Registered, requer sgcWebSockets.

ICalSubscription	TsgcVCLScheduler, TsgcPlannerICalSubscription	Uma URL iCalendar mesclada ao lado de dois itens locais: URL, intervalo de atualização, Atualizar com RefreshAsync, atualização automática com Active, eventos do feed somente leitura por OnItemChanging e IsSubscriptionItem e um log de OnRefreshed, OnError e OnImportWarning. Edição Registered, requer sgcWebSockets.
PlannerFMX	TsgcFMXGantt, TsgcFMXScheduler, TsgcFMXKanban, TsgcPlannerRenderer	As três visões FireMonkey em um objeto de dados para Windows, macOS, iOS e Android: um cartão, uma barra ou um evento movido em uma visão se move nas outras duas. Uma barra de ferramentas com a visão do Scheduler, zoom e ajuste do Gantt, Selected para Done e um tema escuro, e cada visão renderizada sem janela com as opções do seu controle.
HeadlessReport	TsgcPlannerRenderer	Um aplicativo de console sem unit VCL ou FireMonkey, para Windows e Linux64. Ele carrega um projeto com sgcPlannerJSONLoadFromFile, ou grava primeiro o projeto de exemplo como project.json, e salva as páginas scheduler, gantt e kanban como PNG e SVG, após ScrollToTime, ExpandAll e ZoomToFit nas visões que o renderer possui. Uso: `HeadlessReport [project.json] [output folder]`.

Opções de linha de comando

Toda demo tem duas opções usadas pela verificação de build. `/snapshot <file.png>` salva uma imagem da demo e sai: as demos de visão VCL renderizam as suas visões com TsgcPlannerRenderer, DataSetPlanner, ICalendarSync e PlannerFMX salvam a janela (um BMP no Delphi 7), e HeadlessReport grava a página Gantt do projeto de exemplo. `/test [file.txt]`, ou `--test`, executa um autoteste do comportamento principal da demo, grava uma linha PASS ou FAIL por verificação e sai com o número de falhas. As demos de visão conduzem os seus controles com mensagens reais de mouse e de teclado.
As demos VCL compilam com Delphi 7 a Delphi 13, Win32 e Win64. PlannerFMX compila com Delphi XE6 a Delphi 13.

Veja Também

- Início Rápido VCL
- Início Rápido FMX
- Instalação

TsgcPlannerData

Unit: sgcPlanner_Data
TsgcPlannerData são os dados compartilhados do sgcPlanner, um componente não visual. Ele guarda os itens, os recursos, os status, as dependências e o calendário de trabalho, e notifica cada visão vinculada a ele. Um Scheduler, um Gantt, um Kanban, uma origem DataSet, os componentes iCalendar e MS Project e o renderer podem usar o mesmo objeto de dados ao mesmo tempo.

Propriedades

Propriedade	Tipo	Descrição
Items	TsgcPlannerItems	As tarefas, os cartões e os eventos. Veja Items abaixo.
Resources	TsgcPlannerResources	As pessoas ou salas: Id, Name, Color e Initials (calculadas a partir do Name quando vazias). AddResource(aId, aName) e Find(aId).
Statuses	TsgcPlannerStatuses	Os estados do fluxo de trabalho: Id, Caption e Color. São as colunas padrão do Kanban. AddStatus(aId, aCaption, aColor = 0) e Find(aId).
Dependencies	TsgcPlannerDependencies	Os vínculos entre itens: FromId, ToId, Kind (pdkFS padrão, pdkSS, pdkFF, pdkSF) e Lag em minutos de trabalho, negativo para uma antecipação. AddDependency(aFromId, aToId, aKind = pdkFS, aLag = 0).
Calendar	TsgcPlannerCalendar	O calendário de trabalho: WorkDays (padrão segunda a sexta, vazio significa todos os dias), WorkTime (padrão `08:00-12:00,13:00-17:00` , até 4 intervalos, vazio significa o dia inteiro), Holidays (Date, Name) e FirstDayOfWeek (padrão pwdMonday). Métodos IsWorkingDay, IsWorkingTime, WorkingMinutesPerDay, NextWorkingTime, PrevWorkingTime, AddWorkingMinutes e WorkingMinutesBetween.
ProjectStart	TDateTime	O início do projeto para o agendamento automático. 0 usa o Start mais antigo das tarefas. ProjectStartTime é o seu equivalente TsgcPlannerTime.
SelectedId	string	Public. O Id do item selecionado, compartilhado pelas visões que têm SyncSelection.
Version	Int64	Public, somente leitura. Cresce a cada alteração, exceto uma alteração apenas da seleção.
UpdateCount	Integer	Public, somente leitura. A profundidade de BeginUpdate.

Items

Um **TsgcPlannerItem** é uma tarefa no Gantt, um cartão no Kanban e um evento no Scheduler.

Propriedade	Tipo	Descrição
Id	string	A chave única do item. Os pais, os vínculos, os recursos da origem DataSet e os arquivos importados se referem a ela.
Title, Notes	string	O texto do item.
Start, Finish	TDateTime	As datas. Finish é exclusivo. StartTime e FinishTime são os valores exatos TsgcPlannerTime.
AllDay	Boolean	Um evento de dia inteiro, mostrado na faixa de dia inteiro do Scheduler.
Status	string	O Id de um status: a coluna Kanban do cartão. A comparação diferencia maiúsculas de minúsculas.
Progress	Double	Percentual concluído, de 0 a 100.
Color	TsgcUIColor	$AARRGGBB. 0 (padrão) usa a cor do status, depois a do primeiro recurso e depois uma cor da paleta escolhida a partir do Id.
ParentId	string	O Id do item pai. Um item com filhos é um item de resumo.
Tags, Resources	string	Listas separadas por vírgula: tags e Ids de recursos. TagCount, TagAt, ResourceCount e ResourceAt as leem.
Rank	Integer	A posição do cartão na sua coluna Kanban.
Milestone, Manual	Boolean	Um marco é desenhado como um losango. Uma tarefa manual mantém suas datas quando o projeto é agendado.
Constraint, ConstraintDate	TsgcPlannerConstraint, TDateTime	pcASAP (padrão) ou pcSNET, início não antes de ConstraintDate. Usado pelo agendamento automático.
BaselineStart, BaselineFinish	TDateTime	O plano salvo, 0 quando não definido. Para desenhá-lo, a edição Registered.
RecurrenceRule, RecurrenceExceptions	string	Uma RRULE da RFC 5545, como `FREQ=WEEKLY;BYDAY=MO,WE`, e uma lista separada por vírgula de datas ou data e hora excluídas. Expandida na edição Registered.
RecurrenceParentId, RecurrenceDate	string, TDateTime	Definido em uma ocorrência modificada: a série à qual pertence e o início da ocorrência que ela substitui.
IsSummary, Level, Duration, TotalSlack, Critical		Public, somente leitura. Um resumo tem filhos, Level é a profundidade na árvore, Duration os minutos de trabalho no calendário, TotalSlack e Critical o resultado do mecanismo de agendamento.

Métodos

Método	Retorno	Descrição
BeginUpdate, EndUpdate		Agrupam alterações. Os listeners e o OnChange são acionados uma vez após o EndUpdate mais externo.

AddItem(aId = '')	TsgcPlannerItem	Adiciona um item com esse Id.
FindItem(aId), IndexOfItem(aId)	TsgcPlannerItem, Integer	Encontra um item por Id por meio de um índice ordenado. nil ou -1 quando não existe.
DeleteItem(aId)	Boolean	Exclui um item, suas ocorrências modificadas e seus vínculos. Seus filhos passam para o pai dele e a seleção é limpa quando ele estava selecionado. False para um Id desconhecido.
MoveItem(aId, aStatus, aRank)	Boolean	Move um cartão para um status e uma posição, e renumera os ranks das duas colunas. É o que faz um drop no Kanban.
ChildrenOf(aId, var aList)	Integer	Os filhos de um item, ou os itens raiz quando o Id é vazio.
UpdateSummaries		Propaga as datas das folhas para os seus resumos, com o progresso ponderado pela duração de trabalho das folhas.
GetOccurrences(aStart, aFinish, var aList)	Integer	As ocorrências na janela semiaberta [aStart, aFinish) como TsgcPlannerTime, ordenadas pelo início. Na edição Registered um item recorrente gera uma ocorrência por data da sua regra, menos as suas exceções e as suas ocorrências modificadas.
GetRange(out aStart, aFinish)	Boolean	O início mais antigo e o término mais recente dos itens. False quando não há nenhum.
ItemChanging(Sender, aItem, aFields)	Boolean	Aciona OnItemChanging e retorna aAccept. Uma visão personalizada o chama antes de confirmar uma edição.
AddListener, RemoveListener(aListener)		Registram um TsgcPlannerChangeEvent que é avisado de toda alteração. As visões o usam. Um listener não deve alterar os dados.
Clear		Exclui os itens, os recursos, os status e as dependências. O calendário e o ProjectStart são mantidos.
LoadFromJSON, SaveToJSON, LoadFromFile, SaveToFile		Todos os dados como JSON (UTF-8 sem BOM). Um documento que não pode ser lido gera EsgcPlannerError e deixa os dados inalterados.

Eventos

Evento	Tipo	Descrição
OnChange	TsgcPlannerChangeEvent	Após o EndUpdate mais externo de uma alteração, depois dos listeners: aChange com Kinds, ItemId (quando exatamente um item mudou) e Version. O handler pode alterar os dados novamente.
OnItemChanging	TsgcPlannerItemChangingEvent	Antes de uma visão confirmar uma edição feita com o mouse: aItem, aFields (um conjunto de pifStart, pifFinish, pifStatus, pifProgress e os outros campos do item) e var aAccept. Defina aAccept como False para cancelar o drop.

Exemplo: Um calendário de trabalho, uma reunião recorrente e o rastreamento de alterações

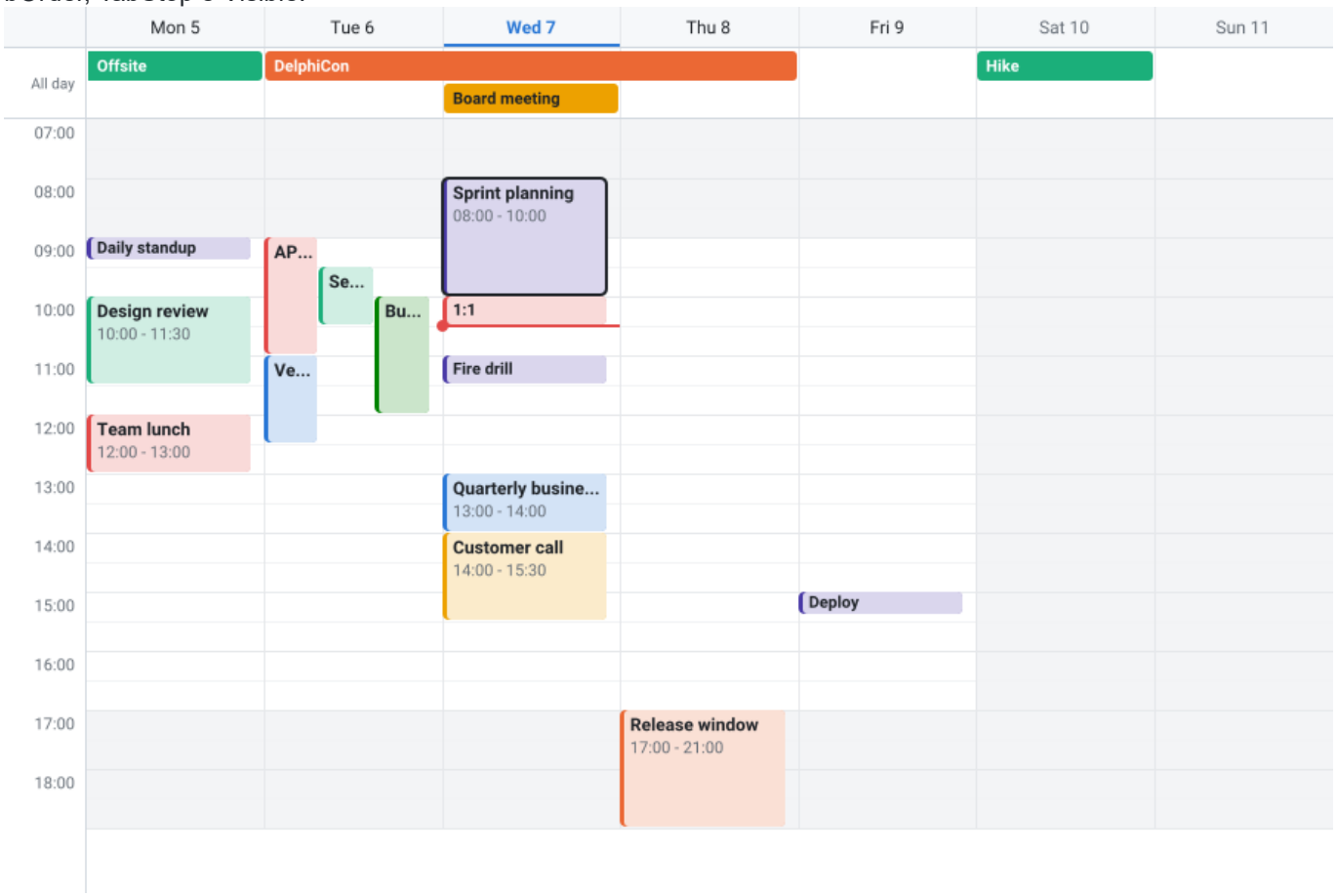
```
uses
  SysUtils, Classes,
  sgcPlanner_DateUtils, sgcPlanner_Data;

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;
```

Veja Também

- Arquitetura
- Início Rápido VCL
- TsgcPlannerSource_DataSet
- Dados Compartilhados
- Calendário de Trabalho
- Recorrência
- Dependências
- Persistência JSON

TsgcVCLScheduler

Unit: sgcPlanner_Scheduler_VCL
TsgcVCLScheduler é o Scheduler da VCL, somente para Windows. Ele mostra os itens de um TsgcPlannerData em uma visão de dia, semana útil, semana, mês ou agenda, com uma faixa de dia inteiro, uma linha no horário atual, eventos sobrepostos lado a lado e, com Options.GroupByResource, uma coluna por recurso. Na edição Registered ele também tem a Timeline de recursos: uma linha por recurso em uma escala de tempo. O usuário cria um evento arrastando sobre espaço livre, move-o e redimensiona-o, e o drop se ajusta a SlotMinutes. Mover para a coluna de outro recurso troca o recurso. Ctrl e a roda do mouse alteram a altura do slot nas visões de dia. Herda de TsgcUIVCLControl e publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop e Visible.



Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados que o controle mostra. Vários controles podem compartilhar um objeto de dados. Quando é nil o controle usa um objeto de dados interno próprio.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
Colors	TsgcPlannerViewColors	Substituições das cores do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. 0 (padrão) mantém a cor do tema.
ReadOnly	Boolean	Desativa todo arrastar e a tecla Delete. Padrão False.

DragOptions	TsgcPlannerDragOptions	Os arrastos que o usuário pode fazer: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos.
SyncSelection	Boolean	Compartilha a seleção com as outras visões dos mesmos dados, por meio de SelectedId dos dados. Padrão True.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
BorderStyle	TBorderStyle	bsSingle (padrão) ou bsNone.
View	TsgcPlannerSchedulerViewKind	psvDay, psvWorkWeek (os dias úteis do calendário), psvWeek (padrão), psvMonth (seis semanas), psvAgenda (uma lista de AgendaDays dias) ou psvTimeline (uma linha por recurso, edição Registered).
Date	TDateTime	A data que a visão mostra. 0 (padrão) segue o relógio.
Options	TsgcPlannerSchedulerOptions	DayStartHour e DayEndHour (padrões 0 e 24), SlotMinutes (30), SlotHeight (20 DIPs), ShowAllDayBand (True), ShowCurrentTime (True), GroupByResource (False), MonthMaxItems (0, quantos couberem), AgendaDays (7) e ResourceRowHeight (40 DIPs, Timeline).
TimeScale	TsgcPlannerTimeScale	O eixo de tempo da Timeline: Zoom (pz5Minutes a pzYears, padrão pzDays), TickWidth, Origin, Snap, ShowNonWorking, MajorFormat e MinorFormat. O zoom também define o período da Timeline: um dia até pzHours, 4 semanas em pzDays, 3 meses em pzWeeks e um ano em pzMonths.
SchedulerView	TsgcPlannerSchedulerView	Public, somente leitura. O modelo de visão, com o layout dos dias, das colunas e das entradas.

Métodos

Método	Retorno	Descrição
GoToday		Define Date como hoje.
Next, Prior		Move Date para o próximo período ou o período anterior da visão.
ScrollToTime(aTime)		Mostra o dia de aTime e rola a grade de horas até o seu horário.
VisibleRange(out aStart, aFinish)		O período que a visão mostra.
ItemAt(X, Y)	TsgcPlannerItem	O item sob um ponto do controle, nil quando não há nenhum.
ScrollToItem(aId)	Boolean	Rola até o item ficar visível. False quando o item não é mostrado.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		A visão como imagem, a mesma imagem de um TsgcPlannerRenderer com os mesmos dados, opções, tamanho e escala. O tamanho é em DIPs, 0 é o tamanho do cliente, e uma escala 0 é a escala do controle. A posição de rolagem é mantida. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Eventos

Evento	Tipo	Descrição
OnItemClick, OnItemDbl-Click	TsgcPlannerIte-mEvent	Um clique ou duplo clique em um item: aItem.
OnItemDelete	TsgcPlannerIte-mAcceptEvent	A tecla Delete no item selecionado: aItem e var aAccept. Defina aAc-cept como False para manter o item. Um item excluído leva seus víncu-los consigo, e seus filhos passam para o pai dele.
OnSelection-Change	TNotifyEvent	O item selecionado mudou, neste controle ou, com SyncSelection, em outra visão dos mesmos dados.
OnItemCreate	TsgcPlannerS-chedulerItem-CreateEvent	Um arrasto sobre espaço livre: aStart, aFinish, aResourceId (a coluna quando agrupado por recurso), var aTitle (padrão "New event") e var aAccept (padrão True). O novo item é selecionado.
OnRecurren-ceEdit	TsgcPlanner-RecurrenceEdi-tEvent	Um drop sobre uma ocorrência de uma série recorrente: aItem (a sé-rie), aOccurrence e var aScope. prsOccurrence (padrão) cria uma ocor-rência modificada, prsSeries desloca a série inteira e prsCancel desfaz o drop. Edição Registered.
OnRecurren-ceDelete	TsgcPlanner-RecurrenceEdi-tEvent	A tecla Delete sobre uma ocorrência de uma série recorrente, antes de qualquer exclusão. prsOccurrence (padrão) adiciona a ocorrência às exceções da série, prsSeries exclui a série após OnItemDelete, prs-Cancel não exclui nada. Edição Registered.
OnDateChan-ge, OnView-Change	TNotifyEvent	Date ou View mudaram, também por um clique em "+N more" na visão de mês, que abre esse dia.

Exemplo: Horário comercial, colunas de recursos e even-tos recorrentes

```
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;
```

Veja Também

- TsgcFMXScheduler
- TsgcPlannerData
- Início Rápido VCL
- Recorrência
- Escala de Tempo e Zoom
- Arrastar e Soltar
- Visões Sincronizadas

TsgcFMXScheduler

Unit: sgcPlanner_Scheduler_FMX
TsgcFMXScheduler é o Scheduler do FireMonkey, a partir do Delphi XE6 (SGC_PLANNER_FMX). Ele tem as mesmas visões, propriedades, métodos e eventos do TsgcVCLScheduler. Ele desenha suas próprias barras de rolagem e trata a entrada por toque como entrada de mouse. Herda de TsgcUIFMXControl e publica as propriedades e eventos de layout usuais do FireMonkey.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados que o controle mostra. Vários controles podem compartilhar um objeto de dados. Quando é nil o controle usa um objeto de dados interno próprio.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
Colors	TsgcPlannerViewColors	Substituições das cores do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. 0 (padrão) mantém a cor do tema.
ReadOnly	Boolean	Desativa todo arrastar e a tecla Delete. Padrão False.
DragOptions	TsgcPlannerDragOptions	Os arrastos que o usuário pode fazer: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos.
SyncSelection	Boolean	Compartilha a seleção com as outras visões dos mesmos dados, por meio de SelectedId dos dados. Padrão True.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
View	TsgcPlannerSchedulerViewKind	psvDay, psvWorkWeek (os dias úteis do calendário), psvWeek (padrão), psvMonth (seis semanas), psvAgenda (uma lista de AgendaDays dias) ou psvTimeline (uma linha por recurso, edição Registered).
Date	TDateTime	A data que a visão mostra. 0 (padrão) segue o relógio.
Options	TsgcPlannerSchedulerOptions	DayStartHour e DayEndHour (padrões 0 e 24), SlotMinutes (30), SlotHeight (20 DIPs), ShowAllDayBand (True), ShowCurrentTime (True), GroupByResource (False), MonthMaxItems (0, quantos couberem), AgendaDays (7) e ResourceRowHeight (40 DIPs, Timeline).
TimeScale	TsgcPlannerTimeScale	O eixo de tempo da Timeline: Zoom (pz5Minutes a pzYears, padrão pzDays), TickWidth, Origin, Snap, ShowNonWorking, MajorFormat e MinorFormat. O zoom também define o período da Timeline: um dia até pzHours, 4 semanas em pzDays, 3 meses em pzWeeks e um ano em pzMonths.
SchedulerView	TsgcPlannerSchedulerView	Public, somente leitura. O modelo de visão, com o layout dos dias, das colunas e das entradas.

Métodos

Método	Retorno	Descrição
GoToday		Define Date como hoje.
Next, Prior		Move Date para o próximo período ou o período anterior da visão.
ScrollToTime(aTime)		Mostra o dia de aTime e rola a grade de horas até o seu horário.
VisibleRange(out aStart, aFinish)		O período que a visão mostra.
ItemAt(X, Y (Single))	TsgcPlannerItem	O item sob um ponto do controle, nil quando não há nenhum.
ScrollToItem(aId)	Boolean	Rola até o item ficar visível. False quando o item não é mostrado.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		A visão como imagem, a mesma imagem de um TsgcPlannerRenderer com os mesmos dados, opções, tamanho e escala. O tamanho é em DIPs, 0 é o tamanho do cliente, e uma escala 0 é a escala do controle. A posição de rolagem é mantida. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Eventos

Evento	Tipo	Descrição
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Um clique ou duplo clique em um item: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	A tecla Delete no item selecionado: aItem e var aAccept. Defina aAccept como False para manter o item. Um item excluído leva seus vínculos consigo, e seus filhos passam para o pai dele.
OnSelectionChange	TNotifyEvent	O item selecionado mudou, neste controle ou, com SyncSelection, em outra visão dos mesmos dados.
OnItemCreate	TsgcPlannerSchedulerItemCreateEvent	Um arrasto sobre espaço livre: aStart, aFinish, aResourceId (a coluna quando agrupado por recurso), var aTitle (padrão "New event") e var aAccept (padrão True). O novo item é selecionado.
OnRecurrenceEdit	TsgcPlannerRecurrenceEditEvent	Um drop sobre uma ocorrência de uma série recorrente: aItem (a série), aOccurrence e var aScope. prsOccurrence (padrão) cria uma ocorrência modificada, prsSeries desloca a série inteira e prsCancel desfaz o drop. Edição Registered.
OnRecurrenceDelete	TsgcPlannerRecurrenceEditEvent	A tecla Delete sobre uma ocorrência de uma série recorrente, antes de qualquer exclusão. prsOccurrence (padrão) adiciona a ocorrência às exceções da série, prsSeries exclui a série após OnItemDelete, prsCancel não exclui nada. Edição Registered.

OnDateChange, OnViewChange	TNotifyEvent	Date ou View mudaram, também por um clique em "+N more" na visão de mês, que abre esse dia.

Exemplo: A Timeline de recursos salva 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;
```

Veja Também

- TsgcVCLScheduler
- Início Rápido FMX
- Recorrência
- Escala de Tempo e Zoom
- Arrastar e Soltar

TsgcVCLGantt

Unit: sgcPlanner_Gantt_VCL
TsgcVCLGantt é o gráfico de Gantt da VCL, somente para Windows. Ele mostra os itens de um TsgcPlannerData como uma árvore de tarefas com números EAP e colunas, e um gráfico de barras em uma escala de tempo de dois níveis: barras com o seu progresso, losangos para os marcos, colchetes para os resumos, o tempo não útil sombreado e uma linha em hoje. O usuário move uma barra, que mantém a sua duração de trabalho, a redimensiona, arrasta o seu progresso, arrasta do fim de uma barra para outra barra para criar um vínculo e abre e fecha os resumos. Um movimento define um início não antes da restrição, a menos que a tarefa seja Manual, e os resumos acompanham os seus filhos. Na edição Registered os vínculos são desenhados como setas, e o Gantt pode agendar o projeto, mostrar o caminho crítico e comparar o plano com uma linha de base. Ctrl e a roda do mouse dão zoom na escala de tempo. Herda de TsgcUIVCLControl e publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop e Visible.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados que o controle mostra. Vários controles podem compartilhar um objeto de dados. Quando é nil o controle usa um objeto de dados interno próprio.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
Colors	TsgcPlannerViewColors	Substituições das cores do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. 0 (padrão) mantém a cor do tema.
ReadOnly	Boolean	Desativa todo arrastar e a tecla Delete. Padrão False.
DragOptions	TsgcPlannerDragOptions	Os arrastos que o usuário pode fazer: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos.
SyncSelection	Boolean	Compartilha a seleção com as outras visões dos mesmos dados, por meio de SelectedId dos dados. Padrão True.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
BorderStyle	TBorderStyle	bsSingle (padrão) ou bsNone.
Columns	TsgcPlannerGanttColumns	As colunas da árvore de tarefas. Cada uma tem um Field (pgcId, pgcWBS, pgcTitle, pgcStart, pgcFinish, pgcDuration, pgcProgress, pgcStatus, pgcResources), uma Caption (vazia usa o título padrão), Width, Visible e Alignment. As colunas padrão são ID, nome da tarefa, início, término, duração e progresso. AddColumn(aField, aWidth = 100, aAlignment = taLeftJustify) e SetDefaults.
TreeWidth	Integer	A largura da árvore de tarefas em DIPs. Padrão 550. O usuário arrasta o divisor para alterá-la.
RowHeight	Integer	A altura de uma linha em DIPs. Padrão 28.

TimeScale	TsgcPlannerTimeScale	O eixo de tempo: Zoom (pz5Minutes, pz15Minutes, pzHours, pzDays padrão, pzWeeks, pzMonths, pzQuarters, pzYears), TickWidth (0 usa o padrão do zoom), Origin, Snap (psNone, psTick padrão, ps5Min, ps15Min, psHour, psDay), ShowNonWorking (True) e MajorFormat e MinorFormat dos dois níveis do cabeçalho.
DurationUnit	TsgcPlannerDurationUnit	A unidade da coluna de duração: pduMinutes, pduHours, pduDays (padrão) ou pduWeeks, contada em tempo de trabalho. O separador decimal é o da locale do Planner.
BarColorMode	TsgcPlannerBarColorMode	pbcItem (padrão) colore uma barra como o item, pbcStatus pelo seu status e pbcResource pelo seu primeiro recurso.
ShowProgress, ShowToday	Boolean	Desenham o progresso dentro das barras e uma linha em hoje. Padrão True.
ShowDependencies	Boolean	Desenha os vínculos como setas em cotovelo. Padrão True. Edição Registered.
ShowCriticalPath, ShowBaseline	Boolean	Desenham as tarefas e os vínculos críticos na cor crítica, e a linha de base sob cada barra. Padrão False. Edição Registered.
AutoSchedule	Boolean	Agenda o projeto novamente após cada edição feita no gráfico. Padrão False. Edição Registered.
GanttView	TsgcPlannerGanttView	Public, somente leitura. O modelo de visão, com as linhas, a geometria das barras e os vínculos.
View	TsgcPlannerView	Public, somente leitura. O modelo de visão que o controle hospeda.

Métodos

Método	Retorno	Descrição
Expand(aId), Collapse(aId)	Boolean	Abre ou fecha um item de resumo.
ExpandAll, CollapseAll		Abre ou fecha todos os itens de resumo.
ZoomIn, ZoomOut	Boolean	Um nível de zoom mais fino ou mais grosso, em torno do centro do gráfico. False no último nível.
ZoomToFit	Boolean	Escolhe o nível de zoom mais fino que mostra todas as linhas e rola até o primeiro início. False sem linhas.
ScrollToTime(aTime)		Rola o gráfico para que aTime fique na sua borda esquerda.
Schedule	Boolean	Executa o mecanismo de agendamento nos dados: tarefas após as suas predecessoras, a partir de ProjectStart, no calendário de trabalho, com a folga e o caminho crítico. Um ciclo nos vínculos aciona OnScheduleError, e as tarefas do ciclo mantêm suas datas. Na Community gera EsgcPlannerError.

SetBaseline		Copia Start e Finish de cada tarefa para BaselineStart e BaselineFinish. Na Community gera EsgcPlannerError.
ItemAt(X, Y)	TsgcPlannerItem	O item sob um ponto do controle, nil quando não há nenhum.
ScrollToItem(aId)	Boolean	Rola até o item ficar visível. False quando o item não é mostrado.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		A visão como imagem, a mesma imagem de um TsgcPlannerRenderer com os mesmos dados, opções, tamanho e escala. O tamanho é em DIPs, 0 é o tamanho do cliente, e uma escala 0 é a escala do controle. A posição de rolagem é mantida. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Eventos

Evento	Tipo	Descrição
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Um clique ou duplo clique em um item: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	A tecla Delete no item selecionado: aItem e var aAccept. Defina aAccept como False para manter o item. Um item excluído leva seus vínculos consigo, e seus filhos passam para o pai dele.
OnSelectionChange	TNotifyEvent	O item selecionado mudou, neste controle ou, com SyncSelection, em outra visão dos mesmos dados.
OnLinkCreating	TsgcPlannerGanttLinkCreatingEvent	Um arrasto do fim de uma barra para outra barra: aFrom, aTo, var aKind (a partir das extremidades escolhidas) e var aAccept. Vínculos para o próprio item, para um ancestral ou um descendente e duplicados são recusados antes do evento. Edição Registered.
OnScheduleError	TsgcPlannerGanttScheduleErrorEvent	O mecanismo encontrou um ciclo: aError e aCycleIds, os Ids dos itens do ciclo. Suas datas são mantidas.
OnExpandChange	TsgcPlannerGanttExpandEvent	Um resumo foi aberto ou fechado: aItem e aExpanded. ExpandAll e CollapseAll o acionam uma vez, com aItem nil.

Exemplo: Colunas EAP, agendamento automático e o caminho crítico

```
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;
```

Veja Também

- TsgcFMXGantt
- TsgcPlannerData
- TsgcPlannerMSProject
- Dependências
- Agendamento Automático
- Caminho Crítico
- Linhas de Base
- Escala de Tempo e Zoom

TsgcFMXGantt

Unit: sgcPlanner_Gantt_FMX
TsgcFMXGantt é o gráfico de Gantt do FireMonkey, a partir do Delphi XE6 (SGC_PLANNER_FMX). Ele tem a mesma árvore, o mesmo gráfico e as mesmas propriedades, métodos e eventos do TsgcVCLGantt. Ele desenha suas próprias barras de rolagem e trata a entrada por toque como entrada de mouse. Herda de TsgcUIFMXControl e publica as propriedades e eventos de layout usuais do FireMonkey.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados que o controle mostra. Vários controles podem compartilhar um objeto de dados. Quando é nil o controle usa um objeto de dados interno próprio.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
Colors	TsgcPlannerViewColors	Substituições das cores do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. 0 (padrão) mantém a cor do tema.
ReadOnly	Boolean	Desativa todo arrastar e a tecla Delete. Padrão False.
DragOptions	TsgcPlannerDragOptions	Os arrastos que o usuário pode fazer: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos.
SyncSelection	Boolean	Compartilha a seleção com as outras visões dos mesmos dados, por meio de SelectedId dos dados. Padrão True.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
Columns	TsgcPlannerGanttColumns	As colunas da árvore de tarefas. Cada uma tem um Field (pgcId, pgcWBS, pgcTitle, pgcStart, pgcFinish, pgcDuration, pgcProgress, pgcStatus, pgcResources), uma Caption (vazia usa o título padrão), Width, Visible e Alignment. As colunas padrão são ID, nome da tarefa, início, término, duração e progresso. AddColumn(aField, aWidth = 100, aAlignment = taLeftJustify) e SetDefaults.
TreeWidth	Integer	A largura da árvore de tarefas em DIPs. Padrão 550. O usuário arrasta o divisor para alterá-la.
RowHeight	Integer	A altura de uma linha em DIPs. Padrão 28.
TimeScale	TsgcPlannerTimeScale	O eixo de tempo: Zoom (pz5Minutes, pz15Minutes, pzHours, pzDays padrão, pzWeeks, pzMonths, pzQuarters, pzYears), TickWidth (0 usa o padrão do zoom), Origin, Snap (psNone, psTick padrão, ps5Min, ps15Min, psHour, psDay), ShowNonWorking (True) e MajorFormat e MinorFormat dos dois níveis do cabeçalho.
DurationUnit	TsgcPlannerDurationUnit	A unidade da coluna de duração: pduMinutes, pduHours, pduDays (padrão) ou pduWeeks, contada em tempo de trabalho. O separador decimal é o da locale do Planner.

BarColorMode	TsgcPlannerBarColorMode	pbcItem (padrão) colore uma barra como o item, pbcStatus pelo seu status e pbcResource pelo seu primeiro recurso.
ShowProgress, ShowToday	Boolean	Desenham o progresso dentro das barras e uma linha em hoje. Padrão True.
ShowDependencies	Boolean	Desenha os vínculos como setas em cotovelo. Padrão True. Edição Registered.
ShowCriticalPath, ShowBaseline	Boolean	Desenham as tarefas e os vínculos críticos na cor crítica, e a linha de base sob cada barra. Padrão False. Edição Registered.
AutoSchedule	Boolean	Agenda o projeto novamente após cada edição feita no gráfico. Padrão False. Edição Registered.
GanttView	TsgcPlannerGanttView	Public, somente leitura. O modelo de visão, com as linhas, a geometria das barras e os vínculos.
View	TsgcPlannerView	Public, somente leitura. O modelo de visão que o controle hospeda.

Métodos

Método	Retorno	Descrição
Expand(aId), Collapse(aId)	Boolean	Abre ou fecha um item de resumo.
ExpandAll, CollapseAll		Abre ou fecha todos os itens de resumo.
ZoomIn, ZoomOut	Boolean	Um nível de zoom mais fino ou mais grosso, em torno do centro do gráfico. False no último nível.
ZoomToFit	Boolean	Escolhe o nível de zoom mais fino que mostra todas as linhas e rola até o primeiro início. False sem linhas.
ScrollToTime(aTime)		Rola o gráfico para que aTime fique na sua borda esquerda.
Schedule	Boolean	Executa o mecanismo de agendamento nos dados: tarefas após as suas predecessoras, a partir de ProjectStart, no calendário de trabalho, com a folga e o caminho crítico. Um ciclo nos vínculos aciona OnScheduleError, e as tarefas do ciclo mantêm suas datas. Na Community gera EsgcPlannerError.
SetBaseline		Copia Start e Finish de cada tarefa para BaselineStart e BaselineFinish. Na Community gera EsgcPlannerError.
ItemAt(X, Y (Single))	TsgcPlannerItem	O item sob um ponto do controle, nil quando não há nenhum.
ScrollToItem(aId)	Boolean	Rola até o item ficar visível. False quando o item não é mostrado.

SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		A visão como imagem, a mesma imagem de um TsgcPlannerRenderer com os mesmos dados, opções, tamanho e escala. O tamanho é em DIPs, 0 é o tamanho do cliente, e uma escala 0 é a escala do controle. A posição de rolagem é mantida. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Eventos

Evento	Tipo	Descrição
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Um clique ou duplo clique em um item: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	A tecla Delete no item selecionado: aItem e var aAccept. Defina aAccept como False para manter o item. Um item excluído leva seus vínculos consigo, e seus filhos passam para o pai dele.
OnSelectionChange	TNotifyEvent	O item selecionado mudou, neste controle ou, com SyncSelection, em outra visão dos mesmos dados.
OnLinkCreating	TsgcPlannerGanttLinkCreatingEvent	Um arrasto do fim de uma barra para outra barra: aFrom, aTo, var aKind (a partir das extremidades escolhidas) e var aAccept. Vínculos para o próprio item, para um ancestral ou um descendente e duplicados são recusados antes do evento. Edição Registered.
OnScheduleError	TsgcPlannerGanttScheduleErrorEvent	O mecanismo encontrou um ciclo: aError e aCycleIds, os Ids dos itens do ciclo. Suas datas são mantidas.
OnExpandChange	TsgcPlannerGanttExpandEvent	Um resumo foi aberto ou fechado: aItem e aExpanded. ExpandAll e CollapseAll o acionam uma vez, com aItem nil.

Exemplo: Horas, um ajuste de 15 minutos e uma fase aberta

```
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;
```

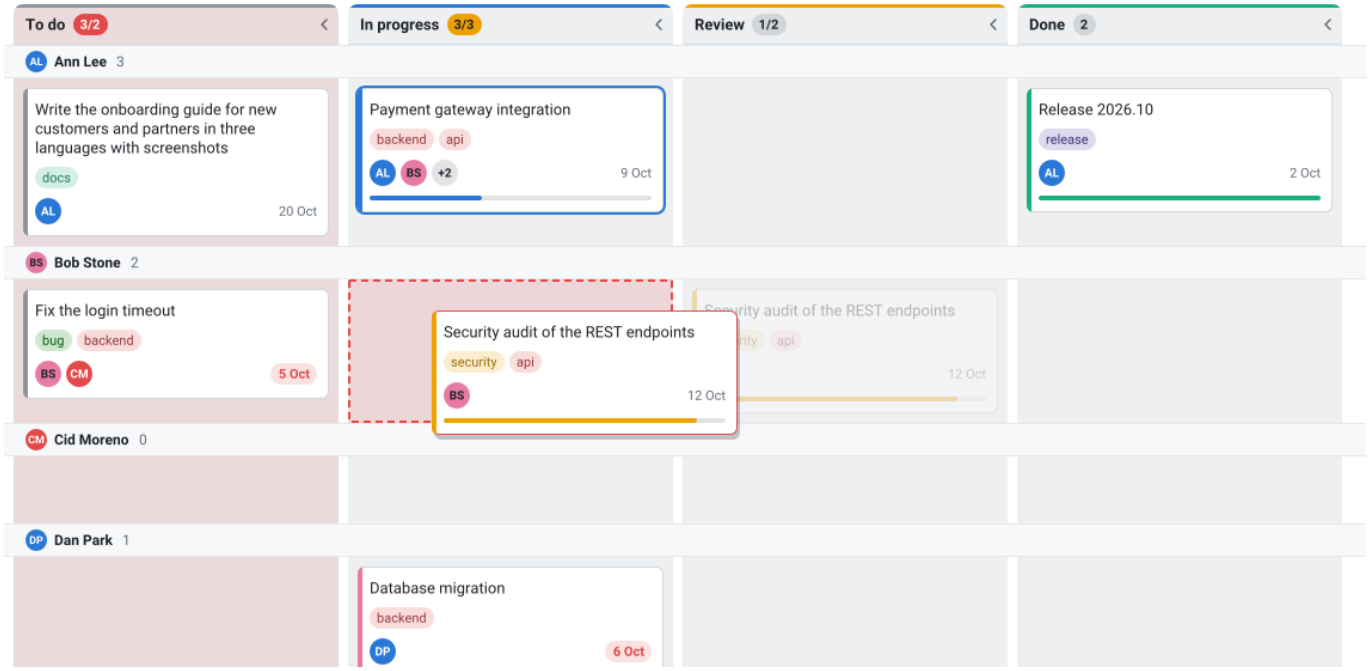
Veja Também

- TsgcVCLGantt

- Início Rápido FMX
- Dependências
- Agendamento Automático
- Caminho Crítico
- Linhas de Base
- Escala de Tempo e Zoom

TsgcVCLKanban

Unit: sgcPlanner_Kanban_VCL
TsgcVCLKanban é o quadro Kanban da VCL, somente para Windows. Ele mostra os itens de um TsgcPlannerData como cartões em um quadro: uma coluna por status, com um contador, um limite de WIP opcional e uma seta de recolher, e na edição Registered raias por recurso, pai, tag ou um evento. Um cartão mostra o seu título em várias linhas, as suas tags, as iniciais dos seus recursos, a sua data de vencimento e o seu progresso. O usuário arrasta um cartão para outra posição, coluna ou raia, e o drop altera o Status e o Rank do item. Itens de resumo e ocorrências modificadas não são mostrados como cartões. Com pdoReorder e sem pdoMove em DragOptions, um arrasto só altera a ordem dos cartões de uma coluna. Herda de TsgcUIVCLControl e publica Align, Anchors, Constraints, Enabled, PopupMenu, ShowHint, ParentShowHint, TabOrder, TabStop e Visible.



Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados que o controle mostra. Vários controles podem compartilhar um objeto de dados. Quando é nil o controle usa um objeto de dados interno próprio.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
Colors	TsgcPlannerViewColors	Substituições das cores do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. 0 (padrão) mantém a cor do tema.
ReadOnly	Boolean	Desativa todo arrastar e a tecla Delete. Padrão False.
DragOptions	TsgcPlannerDragOptions	Os arrastos que o usuário pode fazer: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos.
SyncSelection	Boolean	Compartilha a seleção com as outras visões dos mesmos dados, por meio de SelectedId dos dados. Padrão True.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.

BorderStyle	TBorderStyle	bsSingle (padrão) ou bsNone.
ColumnSource	TsgcPlannerKanbanColumnSource	pkcStatuses (padrão) mostra uma coluna por status dos dados, pkcList mostra apenas a coleção Columns.
Columns	TsgcPlannerKanbanColumns	As colunas: StatusId, Caption e Color (vazios ou 0 usam o status), WIPLimit (0 nenhum), Collapsed e Width (0 padrão). Com pkcStatuses uma entrada apenas acrescenta configurações à coluna do seu status. AddColumn(aStatusId, aCaption = '', aWIPLimit = 0) e Find(aStatusId).
ShowUnknownColumn	Boolean	Mostra uma coluna Outros para os cartões cujo status não tem coluna. Padrão True. Não é possível soltar um cartão nela.
SwimlaneBy	TsgcPlannerSwimlaneBy	pslNone (padrão), pslResource (uma raia por recurso, um drop em outra raia substitui o primeiro recurso), pslParent (uma raia por pai, um drop define ParentId), pslTag (uma raia por tag) ou pslEvent (OnGetSwimlane fornece a raia). Edição Registered.
WIPMode	TsgcPlannerWIPMode	pwmWarn (padrão) apenas colore o contador de uma coluna no ou acima do seu WIPLimit, pwmBlock também recusa um cartão que ultrapassaria o limite. Reordenar dentro de uma coluna é sempre permitido. Edição Registered.
CardOptions	TsgcPlannerKanbanCardOptions	ShowTags, ShowAssignees, ShowDueDate e ShowProgress (padrão True) e MaxTitleLines. A data de vencimento é o Finish do item, mostrada em vermelho quando passou e o cartão não está concluído.
KanbanView	TsgcPlannerKanbanView	Public, somente leitura. O modelo de visão, com as colunas, as raias e os cartões.
View	TsgcPlannerView	Public, somente leitura. O modelo de visão que o controle hospeda.

Métodos

Método	Retorno	Descrição
CollapseColumn(aStatusId), ExpandColumn(aStatusId)	Boolean	Recolhe uma coluna em um trilho estreito ou a abre novamente. O usuário clica na seta do cabeçalho.
SelectNextCard(aKey)	Boolean	Move a seleção com um código de tecla: as setas percorrem os cartões, as colunas e as raias, Home e End vão para o primeiro e o último cartão.
ItemAt(X, Y)	TsgcPlannerItem	O item sob um ponto do controle, nil quando não há nenhum.
ScrollToItem(aId)	Boolean	Rola até o item ficar visível. False quando o item não é mostrado.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		A visão como imagem, a mesma imagem de um TsgcPlannerRenderer com os mesmos dados, opções, tamanho e escala. O tamanho é em DIPs, 0 é o tamanho do cliente, e uma escala 0 é a escala do controle. A posição de rolagem é mantida. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Eventos

Evento	Tipo	Descrição
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Um clique ou duplo clique em um item: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	A tecla Delete no item selecionado: aItem e var aAccept. Defina aAccept como False para manter o item. Um item excluído leva seus vínculos consigo, e seus filhos passam para o pai dele.
OnSelectionChange	TNotifyEvent	O item selecionado mudou, neste controle ou, com SyncSelection, em outra visão dos mesmos dados.
OnCardMoving	TsgcPlannerKanbanCardMovingEvent	Antes de um drop: aItem, aToStatus, aToLane e var aAccept. É acionado antes do OnItemChanging dos dados.
OnCardMoved	TsgcPlannerItemEvent	Depois que um drop foi confirmado: aItem com o seu novo status e rank.
OnGetSwimlane	TsgcPlannerKanbanSwimlaneEvent	Com pslEvent: aItem e var aKey e aCaption, a raia do cartão.
OnColumnCollapse	TsgcPlannerKanbanColumnEvent	Uma coluna foi recolhida ou aberta: aStatusId e aCollapsed.

Exemplo: Um limite de WIP, raias por pessoa e uma regra 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;
```

Veja Também

- TsgcFMXKanban
- TsgcPlannerData
- Início Rápido VCL
- Raias e Limites de WIP
- Arrastar e Soltar
- Temas

TsgcFMXKanban

Unit: sgcPlanner_Kanban_FMX
TsgcFMXKanban é o quadro Kanban do FireMonkey, a partir do Delphi XE6 (SGC_PLANNER_FMX). Ele tem o mesmo quadro e as mesmas propriedades, métodos e eventos do TsgcVCLKanban. Ele desenha suas próprias barras de rolagem e trata a entrada por toque como entrada de mouse, de modo que um dedo arrasta um cartão. Herda de TsgcUIFMXControl e publica as propriedades e eventos de layout usuais do FireMonkey.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados que o controle mostra. Vários controles podem compartilhar um objeto de dados. Quando é nil o controle usa um objeto de dados interno próprio.
Theme	TsgcUITheme	utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
Colors	TsgcPlannerViewColors	Substituições das cores do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. 0 (padrão) mantém a cor do tema.
ReadOnly	Boolean	Desativa todo arrastar e a tecla Delete. Padrão False.
DragOptions	TsgcPlannerDragOptions	Os arrastos que o usuário pode fazer: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos.
SyncSelection	Boolean	Compartilha a seleção com as outras visões dos mesmos dados, por meio de SelectedId dos dados. Padrão True.
MaxFPS	Integer	O maior número de quadros por segundo que o controle desenha.
ColumnSource	TsgcPlannerKanbanColumnSource	pkcStatuses (padrão) mostra uma coluna por status dos dados, pkcList mostra apenas a coleção Columns.
Columns	TsgcPlannerKanbanColumns	As colunas: StatusId, Caption e Color (vazios ou 0 usam o status), WIPLimit (0 nenhum), Collapsed e Width (0 padrão). Com pkcStatuses uma entrada apenas acrescenta configurações à coluna do seu status. AddColumn(aStatusId, aCaption = '', aWIPLimit = 0) e Find(aStatusId).
ShowUnknownColumn	Boolean	Mostra uma coluna Outros para os cartões cujo status não tem coluna. Padrão True. Não é possível soltar um cartão nela.
SwimlaneBy	TsgcPlannerSwimlaneBy	pslNone (padrão), pslResource (uma raia por recurso, um drop em outra raia substitui o primeiro recurso), pslParent (uma raia por pai, um drop define ParentId), pslTag (uma raia por tag) ou pslEvent (OnGetSwimlane fornece a raia). Edição Registered.
WIPMode	TsgcPlannerWIPMode	pwmWarn (padrão) apenas colore o contador de uma coluna no ou acima do seu WIPLimit, pwmBlock também recusa um cartão que ultrapassaria o limite. Reordenar dentro de uma coluna é sempre permitido. Edição Registered.

CardOptions	TsgcPlannerKanbanCardOptions	ShowTags, ShowAssignees, ShowDueDate e ShowProgress (padrão True) e MaxTitleLines. A data de vencimento é o Finish do item, mostrada em vermelho quando passou e o cartão não está concluído.
KanbanView	TsgcPlannerKanbanView	Public, somente leitura. O modelo de visão, com as colunas, as raias e os cartões.
View	TsgcPlannerView	Public, somente leitura. O modelo de visão que o controle hospeda.

Métodos

Método	Retorno	Descrição
CollapseColumn(aStatusId), ExpandColumn(aStatusId)	Boolean	Recolhe uma coluna em um trilho estreito ou a abre novamente. O usuário clica na seta do cabeçalho.
SelectNextCard(aKey)	Boolean	Move a seleção com um código de tecla: as setas percorrem os cartões, as colunas e as raias, Home e End vão para o primeiro e o último cartão.
ItemAt(X, Y (Single))	TsgcPlannerItem	O item sob um ponto do controle, nil quando não há nenhum.
ScrollToItem(aId)	Boolean	Rola até o item ficar visível. False quando o item não é mostrado.
SaveToPNG, SaveToSVG(aFileName or aStream, aWidth = 0, aHeight = 0, aScale = 0)		A visão como imagem, a mesma imagem de um TsgcPlannerRenderer com os mesmos dados, opções, tamanho e escala. O tamanho é em DIPs, 0 é o tamanho do cliente, e uma escala 0 é a escala do controle. A posição de rolagem é mantida. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Eventos

Evento	Tipo	Descrição
OnItemClick, OnItemDblClick	TsgcPlannerItemEvent	Um clique ou duplo clique em um item: aItem.
OnItemDelete	TsgcPlannerItemAcceptEvent	A tecla Delete no item selecionado: aItem e var aAccept. Defina aAccept como False para manter o item. Um item excluído leva seus vínculos consigo, e seus filhos passam para o pai dele.
OnSelectionChange	TNotifyEvent	O item selecionado mudou, neste controle ou, com SyncSelection, em outra visão dos mesmos dados.
OnCardMoving	TsgcPlannerKanbanCardMovingEvent	Antes de um drop: aItem, aToStatus, aToLane e var aAccept. É acionado antes do OnItemChanging dos dados.
OnCardMoved	TsgcPlannerItemEvent	Depois que um drop foi confirmado: aItem com o seu novo status e rank.

OnGetSwimlane	TsgcPlannerKanbanSwimlaneEvent	Com pslEvent: aItem e var aKey e aCaption, a raia do cartão.
OnColumnCollapse	TsgcPlannerKanbanColumnEvent	Uma coluna foi recolhida ou aberta: aStatusId e aCollapsed.

Exemplo: Uma lista fixa de colunas e uma coluna Done recolhida

```
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;
```

Veja Também

- TsgcVCLKanban
- Início Rápido FMX
- Raias e Limites de WIP
- Arrastar e Soltar

TsgcPlannerSource_DataSet

Unit: sgcPlanner_Source_DataSet
TsgcPlannerSource_DataSet vincula um TsgcPlannerData a TDataSets: DataSet guarda os itens, ResourceDataSet os recursos e DependencyDataSet as dependências. Precisa apenas da unit DB, de modo que funciona com qualquer dataset, em VCL, FireMonkey ou em um serviço. Faz parte de todas as edições.

Carregamento

- Active True carrega todo DataSet aberto, e abrir um DataSet enquanto Active o carrega. Um carregamento substitui os itens, os recursos e as dependências em uma única atualização, de modo que cada visão vinculada recebe uma notificação. O registro atual de cada DataSet é restaurado depois.
- As datas vêm de campos de data e hora, de campos numéricos ou de texto ISO 8601. Os booleanos vêm de campos booleanos ou numéricos ou de texto (1, T, True, Y, Yes). As cores vêm de campos numéricos (ARGB) ou de texto (#RRGGBB, #AARRGGBB, $AARRGGBB). Um nulo dá o valor padrão, e um Finish nulo dá Finish = Start, ou a próxima meia-noite para um item de dia inteiro.
- Sem um DependencyDataSet, DependsOnField guarda os Ids separados por vírgula das predecessoras do item, lidos como vínculos término a início.

Atualizações ao vivo e gravação de volta

- Com LiveUpdates, uma edição ou inserção postada atualiza ou adiciona somente esse item, e uma mudança na contagem de registros (uma exclusão, um filtro, um refresh) sincroniza os itens por Id. Rolar e edições não postadas não alteram nada.
- Com WriteBack, uma edição em uma visão ou no código é gravada no DataSet: o registro é encontrado com Locate no campo Id, os campos mapeados alterados são atribuídos e o registro é postado de imediato. Um novo item acrescenta um registro e um item excluído exclui o seu registro. Recursos nunca são gravados, nem campos calculados, de lookup e somente leitura. Nada é gravado enquanto o DataSet está em estado de edição ou inserção, e as alterações pendentes são gravadas quando ele volta ao estado de navegação.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados a preencher.
DataSet	TDataSet	Os itens, um registro por item.
ItemFields	TsgcPlannerSourceItemFields	O campo de cada valor do item: IdField, TitleField, StartField, FinishField, AllDayField, StatusField, ProgressField, ColorField, ParentField, TagsField, ResourcesField, RankField, NotesField, RecurrenceField e DependsOnField. Os padrões são Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, RecurrenceRule e DependsOn. Um nome vazio ou não encontrado no DataSet é ignorado.
ResourceDataSet, ResourceFields	TDataSet, TsgcPlannerSourceResourceFields	Os recursos: IdField, NameField, ColorField e InitialsField, padrões Id, Name, Color e Initials. Somente leitura.
DependencyDataSet, De-	TDataSet, TsgcPlannerSourceDe-	Os vínculos: FromField, ToField, KindField (FS, SS, FF, SF ou o ordinal) e LagField (minutos de trabalho), padrões FromId, ToId, Kind e Lag.

pendency-Fields	pendencyFi-elds	
Active	Boolean	Carrega os DataSets abertos e os acompanha. Padrão False.
LiveUpda-tes	Boolean	Acompanha as postagens, inserções, exclusões e refreshes dos DataSets. Padrão True.
WriteBack	Boolean	Grava as alterações dos dados em DataSet e DependencyDataSet. Padrão True.

Métodos

Método	Retorno	Descrição
Load		Carrega agora todo DataSet aberto, com Active ou não. Nada acontece quando Data é nil.
Refresh		Carrega novamente quando Active.

Eventos

Evento	Tipo	Descrição
OnLoa-dItem	TsgcPlanner-SourceLoadIte-mEvent	Depois que os campos mapeados de um registro foram copiados para o seu item: aDataSet, nesse registro, e aItem, para que a aplicação possa preen-cher outras propriedades do item.

Exemplo: Tarefas e pessoas a partir de duas tabelas

```
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;
```

Veja Também

- TsgcPlannerData
- Início Rápido VCL
- Vínculo com DataSet

TsgcPlannerICalendar

Unit: sgcPlanner_ICal
TsgcPlannerICalendar importa e exporta os VEVENTs de arquivos iCalendar (RFC 5545), o formato do Google Agenda, do Outlook e do Apple Calendar, para dentro e a partir de um TsgcPlannerData. Faz parte da edição Registered (SGC_PLANNER_ADVANCED).

- O Id de um item é o UID do seu evento. Um evento com um RECURRENCE-ID torna-se uma ocorrência modificada da sua série.
- Horários em UTC (Z) e com um TZID são convertidos para o relógio local, com o VTIMEZONE do arquivo. Um TZID desconhecido gera um aviso e é lido como horário local.
- STATUS é mapeado para o Status do item, RESOURCES para os seus recursos (recursos ausentes são criados), RELATED-TO com RELTYPE=PARENT para ParentId, e RRULE e EXDATE para a recorrência. Os outros campos do item viajam em propriedades X-SGC, de modo que uma exportação seguida de uma importação dá os mesmos dados. As dependências não são exportadas.
- Um texto sem BEGIN:VCALENDAR gera EsgcPlannerError e deixa os dados inalterados. Uma importação é uma única atualização, de modo que as visões recebem uma notificação.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados para importar ou exportar.
ImportMode	TsgcPlannerICalImportMode	pimReplace (padrão) exclui primeiro os itens e as dependências e mantém os recursos, os status, o calendário e o ProjectStart. pimMerge atualiza os itens com o mesmo Id e adiciona os outros.
ExportUTC	Boolean	Grava em UTC os horários dos eventos que não se repetem. Séries e ocorrências modificadas são sempre gravadas como horários locais flutuantes, que continuam corretos nas mudanças de horário de verão. Padrão True.
ProductId	string	O PRODID do calendário exportado. Padrão `-//eSeGeCe//sgcPlanner//EN`.
WarningCount	Integer	Public, somente leitura. Os avisos da última importação.

Métodos

Método	Retorno	Descrição
ImportFromFile, ImportFromStream, ImportFromString		Lê um calendário. Arquivos e streams são UTF-8, com ou sem BOM. UTF-8 inválido é lido como Latin-1, com um aviso.
ExportToFile, ExportToStream		Grava os dados como um calendário, em UTF-8.
ExportToString	string	Os dados como texto iCalendar.

Eventos

Evento	Tipo	Descrição
OnImportWarning	TsgcPlannerICalWarningEvent	Uma parte do arquivo que foi ignorada ou lida de forma aproximada: aLine e aMessage, por exemplo uma RRULE fora do subconjunto suportado, um RDATE ou um fuso horário desconhecido.

Exemplo: Mesclar um calendário e gravá-lo de volta

```
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;
```

Veja Também

- TsgcPlannerMSProject
- TsgcPlannerData
- Edições
- iCalendar
- Recorrência

TsgcPlannerMSProject

Unit: sgcPlanner_MSProject
TsgcPlannerMSProject importa e exporta arquivos XML do Microsoft Project (MSPDI), o formato XML que o Microsoft Project salva e abre, para dentro e a partir de um TsgcPlannerData. Faz parte da edição Registered (SGC_PLANNER_ADVANCED).

- As tarefas viram itens: o Id é o UID, a estrutura de tópicos dá o ParentId, e as datas, o progresso, o marco, o indicador manual, a restrição e a linha de base seguem o arquivo. Os vínculos de predecessoras viram dependências com o seu tipo e a sua latência, e as atribuições dão os recursos de cada tarefa.
- O calendário do projeto dá os dias úteis, o horário de trabalho e os feriados, e a data de início do projeto dá o ProjectStart.
- No sgcPlanner existem apenas as restrições ASAP e início não antes de. Deve começar em é lida como início não antes de, e os outros tipos de restrição como ASAP, cada um com um aviso.
- A exportação grava um calendário Standard, as tarefas na ordem da estrutura de tópicos, os vínculos e os recursos. Status, tags, cor, rank, dia inteiro e recorrência não são gravados.
- Um documento que não está bem formado gera EsgcPlannerError com a linha e deixa os dados inalterados.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados para importar ou exportar.
ImportMode	TsgcPlannerMSProjectImportMode	pmiReplace (padrão) exclui os itens e os vínculos e toma o calendário e o início do projeto do arquivo. Recursos e status são mantidos, e os recursos do arquivo são adicionados. pmiMerge mantém o calendário e o ProjectStart, atualiza os itens com o mesmo Id e adiciona os outros.
WarningCount	Integer	Public, somente leitura. Os avisos da última importação.

Métodos

Método	Retorno	Descrição
ImportFromFile, ImportFromStream, ImportFromString		Lê um projeto. Arquivos e streams são UTF-8, com ou sem BOM.
ExportToFile, ExportToStream		Grava os dados como um projeto.
ExportToString	string	Os dados como texto MSPDI.

Eventos

Evento	Tipo	Descrição
OnImportWarning	TsgcPlannerMSProjectWarningEvent	Uma parte do arquivo que foi lida de forma aproximada: aLine e aMessage.

Exemplo: Abrir e salvar um arquivo do 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;
```

Veja Também

- TsgcVCLGantt
- TsgcPlannerICalendar
- TsgcPlannerData
- XML do MS Project
- Linhas de Base

TsgcPlannerRenderer

Unit: sgcPlanner_Renderer
TsgcPlannerRenderer desenha uma visão Scheduler, Gantt ou Kanban de um TsgcPlannerData sem janela, em qualquer TsgcUICanvas, em um PNG ou em um SVG, de modo que um servidor, um relatório ou um aplicativo de console dá a mesma imagem que um formulário mostra. Ele possui um modelo de visão de cada tipo, as mesmas classes que os controles criam, de modo que a saída coincide pixel a pixel com os controles. É neutro em relação ao framework e roda no Windows e no Linux64, em todas as edições.

Propriedades

Propriedade	Tipo	Descrição
Data	TsgcPlannerData	Os dados a desenhar. A seleção é SelectedId dos dados.
View	TsgcPlannerRendererViewKind	prvScheduler (padrão), prvGantt ou prvKanban.
Date	TDateTime	A data do Scheduler. 0 (padrão) segue o relógio.
Width, Height	Integer	O tamanho em DIPs. Padrão 800 x 600.
Scale	Double	O fator HiDPI. A saída tem Round(Width * Scale) x Round(Height * Scale) pixels. Padrão 1.
Theme, Colors	TsgcUITheme, TsgcPlannerViewColors	Como nos controles.
FontFile, FontFileBold	string	Arquivos TrueType com os quais medir e desenhar o texto. Vazio (padrão) usa a fonte embutida. O Kanban quebra os títulos dos cartões com FontFile.
SchedulerOptions	TsgcPlannerRendererSchedulerOptions	View, Options e TimeScale do Scheduler, como em TsgcVCLScheduler.
GanttOptions	TsgcPlannerRendererGanttOptions	Columns, TreeWidth, RowHeight, TimeScale, DurationUnit, BarColorMode, ShowProgress, ShowToday, ShowDependencies, ShowCriticalPath e ShowBaseline, como em TsgcVCLGantt.
KanbanOptions	TsgcPlannerRendererKanbanOptions	ColumnSource, Columns, ShowUnknownColumn, SwimlaneBy, WIPMode e CardOptions, como em TsgcVCLKanban.
SchedulerView, GanttView, KanbanView		Public, somente leitura. Os modelos de visão, para o que as opções não cobrem, como ExpandAll, Collapse ou ScrollToTime. O tamanho deles segue Width e Height, de modo que uma rolagem definida antes da renderização é mantida.
ActiveView	TsgcPlannerView	Public, somente leitura. O modelo de visão de View.

Cada classe de opções também copia as opções de um modelo de visão, de modo que `GanttOptions.Assign(VCLGantt1.GanttView)` renderiza o que esse controle mostra.

Métodos

Método	Retorno	Descrição
RenderTo(aCanvas)		Desenha a visão em um TsgcUICanvas de DeviceWidth x DeviceHeight pixels.
SaveToPNG(aFileName or aStream)		A visão como PNG.
SaveToSVG(aFileName or aStream)		A visão como SVG.
DeviceWidth, DeviceHeight	Integer	O tamanho da saída em pixels.

Eventos

TsgcPlannerRenderer não tem eventos. Os eventos OnItemChanging e OnChange dos dados são acionados normalmente quando os dados mudam.

Exemplo: Um Gantt como SVG e um mês escuro 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;
```

Veja Também

- Arquitetura
- TsgcPlannerData
- TsgcVCLGantt
- Renderização sem Janela

- Temas

Dados Compartilhados

Toda visão do sgcPlanner mostra um TsgcPlannerData. O objeto de dados guarda todo o planner: os itens, os recursos, os status, as dependências, o calendário de trabalho, o ProjectStart e a seleção. As visões não guardam cópia dele, elas o leem, de modo que o Gantt, o quadro Kanban e o Scheduler podem mostrar os mesmos itens ao mesmo tempo, e uma alteração feita em um deles, ou no código, chega a todos.

O que os dados guardam

Parte	Conteúdo
Items	TsgcPlannerItem: Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, Milestone, Manual, Constraint, ConstraintDate, as datas de linha de base e a recorrência. Um item é uma tarefa no Gantt, um cartão no quadro e um evento no Scheduler.
Resources	Pessoas ou salas: Id, Name, Color e Initials. Um item nomeia os seus recursos pelo Id na sua lista Resources separada por vírgula.
Statuses	O fluxo de trabalho: Id, Caption e Color. São as colunas do quadro e podem colorir as barras e os eventos.
Dependencies	Os vínculos entre itens: FromId, ToId, Kind e Lag. Veja Dependências.
Calendar	Os dias úteis, o horário de trabalho e os feriados. Veja Calendário de Trabalho.
ProjectStart, SelectedId	O início do projeto para o mecanismo de agendamento e o item selecionado de cada visão com SyncSelection.

Um item é encontrado pelo seu Id com FindItem e IndexOfItem. ParentId constrói a árvore de tarefas: ChildrenOf('') retorna as raízes, e um item de resumo obtém as suas datas e o seu progresso das suas folhas com UpdateSummaries. DeleteItem remove os vínculos do item e as suas ocorrências modificadas, e move os seus filhos para o seu pai. MoveItem altera o status e o rank de um cartão em uma única chamada.

Notificações de alteração

Todo setter notifica. Entre BeginUpdate e EndUpdate as alterações são reunidas, e após o EndUpdate mais externo cada listener recebe um TsgcPlannerChange:

Campo	Significado
Kinds	O conjunto de alterações: pckItems (um campo de um item), pckStructure (um item adicionado ou excluído, um Id, um ParentId ou a ordem mudou), pckResources, pckStatuses, pckDependencies, pckCalendar, pckSelection e pckSchedule (ProjectStart, ou os resultados TotalSlack e Critical do mecanismo).
ItemId	O Id do item quando exatamente um item mudou, caso contrário vazio.
Version	Um número que cresce a cada alteração, exceto uma alteração apenas da seleção. Duas versões iguais significam os mesmos dados.

As visões são listeners. Uma alteração apenas da seleção redesenha somente o overlay delas, qualquer outra alteração refaz o layout no próximo quadro. O seu próprio código escuta com o evento OnChange ou com AddListener, que aceita qualquer número de handlers; chame RemoveListener antes que o objeto dono de um handler seja liberado. Um handler que altera os dados novamente inicia mais uma rodada de notificações, e após 8 rodadas EsgcPlannerError é gerado, de modo que dois handlers que se alteram sem parar não travam a aplicação.

Exemplo

```
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;
```

Veja Também

- Visões Sincronizadas
- TsgcPlannerData
- Arquitetura

Visões Sincronizadas

Vincule duas ou mais visões ao mesmo TsgcPlannerData e elas permanecem em sintonia. Um cartão arrastado para Done no quadro é desenhado com a cor de Done no Gantt e no Scheduler na hora, uma barra movida no Gantt move o evento no Scheduler, e um evento criado no Scheduler vira um cartão. Cada edição é confirmada em uma única atualização, de modo que cada visão se redesenha uma vez por drop.

Seleção compartilhada

Com SyncSelection, que é True por padrão, a seleção de uma visão é SelectedId dos dados. Clique em uma barra e o mesmo item é selecionado no quadro e no Scheduler. Com SyncSelection False a visão mantém uma seleção própria. Defina SelectedId no código para selecionar um item em todas as visões sincronizadas. OnSelection-Change é acionado para uma seleção feita no controle ou, com SyncSelection, em outra visão dos mesmos dados, que é o lugar para rolar as outras visões até o item com ScrollToItem.

Visões em vários formulários

Coloque o TsgcPlannerData em um data module e atribua-o às visões de cada formulário. As visões de um formulário que não está visível também são atualizadas e desenham o estado atual quando são exibidas.

A mesma imagem fora do formulário

As opções de uma visão podem ser copiadas para um TsgcPlannerRenderer com Assign, por exemplo `Renderer.GanttOptions.Assign(VCLGantt1.GanttView)`, de modo que um relatório mostra o que o formulário mostra. Os controles também se salvam como PNG e SVG. Veja Renderização sem Janela.

Exemplo

```
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;
```

Veja Também

- Dados Compartilhados
- Arrastar e Soltar
- Início Rápido VCL

Calendário de Trabalho

TsgcPlannerData.Calendar é o calendário de trabalho do planner: os dias da semana trabalhados, o horário de trabalho de um dia e os feriados. As durações, as latências, os movimentos do Gantt e o mecanismo de agendamento contam minutos de trabalho nele, de modo que uma tarefa de dois dias úteis que começa em uma sexta-feira termina na segunda-feira à noite.

Propriedades

Propriedade	Descrição
WorkDays	Os dias da semana trabalhados, um conjunto de pwdMonday a pwdSunday. Padrão segunda a sexta. Um conjunto vazio significa que todo dia é dia útil.
WorkTime	O horário de trabalho de um dia útil como até 4 intervalos, `hh:nn-hh:nn` separados por vírgula. Padrão `08:00-12:00,13:00-17:00`. Um texto vazio significa o dia inteiro. Um texto inválido gera EsgcPlannerError, um válido é armazenado na forma canônica.
Holidays	Datas que não são trabalhadas, cada uma com um Name. AddHoliday(Date, Name) adiciona uma. Um feriado em um dia de folga é ignorado.
FirstDayOfWeek	O primeiro dia de uma semana no Scheduler e nas marcas de semana da escala de tempo. Padrão pwdMonday.

Aritmética do tempo de trabalho

Os métodos recebem e retornam TsgcPlannerTime, a contagem exata de minutos Int64 do modelo. sgcPlannerTimeOf e sgcPlannerDateTimeOf convertem de e para TDateTime.

Método	Retorno
IsWorkingDay(aTime), IsWorkingTime(aTime)	Se o dia, ou o minuto, é trabalhado.
WorkingMinutesPerDay	Os minutos de trabalho de um dia útil, 480 com o WorkTime padrão.
NextWorkingTime(aTime), PrevWorkingTime(aTime)	O primeiro minuto de trabalho em ou após aTime, ou o último antes dele.
AddWorkingMinutes(aTime, aMinutes)	O horário após aMinutes minutos de trabalho. Uma contagem positiva retorna o fim do último minuto de trabalho, de modo que segunda-feira 08:00 mais 480 é segunda-feira 17:00, não terça-feira 08:00. Uma contagem negativa volta no tempo e retorna o início do último minuto.
WorkingMinutesBetween(aStart, aFinish)	Os minutos de trabalho de [aStart, aFinish), negativos quando aFinish é anterior a aStart.
GetNonWorkingIntervals(aStart, aFinish, aList)	As noites, as folgas e os feriados de um período, como a escala de tempo os desenha.

Onde o calendário é usado

- O Gantt mostra durações em tempo de trabalho (DurationUnit), mantém a duração de trabalho de uma barra que é movida e sombreia o tempo não útil quando TimeScale.ShowNonWorking é True.
- O Lag de uma dependência é em minutos de trabalho, e o mecanismo de agendamento coloca cada tarefa em tempo de trabalho. Veja Agendamento Automático.
- UpdateSummaries pondera o progresso de cada folha pela sua duração de trabalho.
- A visão de semana útil do Scheduler mostra os dias de WorkDays.

Uma alteração do calendário notifica pckCalendar, de modo que as visões se redesenham, e é salva com JSON e XML do MS Project.

Exemplo

```
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;
```

Veja Também

- Agendamento Automático
- Dependências
- Escala de Tempo e Zoom
- TsgcPlannerData

Recorrência

Edição: Registered. Na edição Community um item recorrente é mostrado uma vez, no seu próprio Start, e a sua regra é mantida e salva. Veja Recursos por Edição.
Um item com uma RecurrenceRule é uma série: ele se repete nas datas de uma RRULE do iCalendar (RFC 5545). O Scheduler mostra cada ocorrência, GetOccurrences dos dados as retorna para qualquer período, e os arquivos iCalendar as importam e exportam.

A regra

RecurrenceRule guarda o texto da regra, com ou sem o prefixo `RRULE:`. O Start e o Finish do item são a primeira ocorrência, e toda ocorrência tem a mesma duração.

Parte	Suporte
FREQ	DAILY, WEEKLY, MONTHLY ou YEARLY.
INTERVAL	A cada n períodos, por exemplo a cada duas semanas.
COUNT, UNTIL	O número de ocorrências, ou a última data. UNTIL aceita uma data, uma data e hora local ou uma data e hora UTC.
BYDAY	Dias da semana, por exemplo MO,WE,FR. Em regras MONTHLY e YEARLY um dia pode ter um ordinal de -5 a 5, de modo que 2TU é a segunda terça-feira e -1FR a última sexta-feira. Com DAILY ele filtra os dias.
BYMONTH, BYMONTHDAY	Meses do ano e um dia do mês (MONTHLY e YEARLY). Uma regra YEARLY com BYDAY precisa de BYMONTH.
WKST	O primeiro dia da semana da regra.

A primeira ocorrência é sempre o Start do item e conta para COUNT. Uma regra inválida faz o item aparecer uma vez. Para verificar uma regra digitada por um usuário, chame TryParse de TsgcPlannerRecurrence, que retorna False e o motivo.

Exceções e ocorrências modificadas

- **RecurrenceExceptions** é uma lista separada por vírgula de datas ou data e hora ISO. Uma data remove as ocorrências desse dia inteiro, uma data e hora remove a ocorrência que começa nesse minuto. Uma ocorrência removida continua contando para COUNT.
- Uma **ocorrência modificada** é um item próprio com RecurrenceParentId definido como o Id da série e RecurrenceDate definido como o início da ocorrência que ele substitui. Pode ter outros horários, outro título ou outros recursos, e a ocorrência gerada dessa data deixa de ser mostrada.
- DeleteItem de uma série também exclui as suas ocorrências modificadas.

Editando no Scheduler

Um drop sobre uma ocorrência gerada aciona OnRecurrenceEdit com a série, a ocorrência e um escopo: prsOccurrence (o padrão) cria uma ocorrência modificada, prsSeries move a série inteira na mesma quantidade e prsCancel cancela o drop. A tecla Delete sobre uma ocorrência aciona OnRecurrenceDelete com os mesmos escopos: prsOccurrence (o padrão) remove somente essa ocorrência, acrescentando a sua data às exceções da série e excluindo-a quando é uma ocorrência modificada, e prsSeries exclui a série inteira.
O mecanismo de agendamento mantém as datas de uma série e de suas ocorrências modificadas. O quadro Kanban não mostra ocorrências modificadas como cartões.

Exemplo

```
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;
```

Veja Também

- TsgcVCLScheduler
- iCalendar
- TsgcPlannerData

- Calendário de Trabalho

Escala de Tempo e Zoom

O Gantt e a Timeline de recursos do Scheduler desenham o tempo em um eixo horizontal descrito por um TsgcPlannerTimeScale, a propriedade TimeScale do controle (GanttOptions.TimeScale e SchedulerOptions.TimeScale no renderer). Ele mapeia o tempo em pixels, desenha dois níveis de marcas com rótulos, sombreia o tempo não útil e ajusta os arrastos.

Níveis de zoom

Zoom	Marca menor	Marca maior	Largura padrão	Rótulos (menor, maior)
pz5Minutes	5 minutos	hora	20	`nn` , `ddd d mmm t`
pz15Minutes	15 minutos	hora	30	`nn` , `ddd d mmm t`
pzHours	hora	dia	40	`hh` , `dddd d mmmm yyyy`
pzDays (padrão)	dia	semana	32	`d` , `d mmm yyyy`
pzWeeks	semana	mês	56	`d mmm` , `mmmm yyyy`
pzMonths	mês	ano	60	`mmm` , `yyyy`
pzQuarters	trimestre	ano	60	`"Q"q` , `yyyy`
pzYears	ano	década	60	`yyyy` , `yyyy`

O eixo é linear no tempo. Nos zooms grosseiros um mês conta como 30 dias, um trimestre como 91 e um ano como 365, de modo que as marcas de meses de duração diferente têm larguras ligeiramente diferentes.

Propriedades

Propriedade	Descrição
Zoom	O nível de zoom. Padrão pzDays.
TickWidth	A largura de uma marca menor em DIPs. 0 (padrão) é a largura do zoom na tabela acima.
Origin	O tempo em X = 0 do eixo.
Snap	O passo de um movimento ou redimensionamento: psTick (padrão, a marca menor até pzDays e dias inteiros em zooms mais grosseiros), psNone, ps5Min, ps15Min, psHour ou psDay.
ShowNonWorking	Sombreia as noites, as folgas e os feriados do calendário de trabalho. Padrão True.
MajorFormat, MinorFormat	Os rótulos dos dois níveis. Vazio usa os formatos da tabela acima.

Os tokens de formato não diferenciam maiúsculas de minúsculas: `yyyy` e `yy` para o ano, `q` para o trimestre, `m` a `mmmm` para o mês, `d` a `dddd` para o dia, `w` e `ww` para a semana ISO 8601, `h` e `hh` para a hora de 0 a 23, `n` e `nn` para os minutos e `t` para a hora no relógio da locale. Texto entre aspas simples ou duplas é copiado como está. Os nomes vêm de sgcPlannerLocale.

Dando zoom

- ZoomIn e ZoomOut do Gantt vão um nível mais fino ou mais grosso e retornam False no fim do intervalo. ZoomToFit escolhe o zoom que mostra o projeto inteiro na largura do gráfico.
- Ctrl e a roda do mouse dão zoom no Gantt e na Timeline na posição do mouse, de modo que o tempo sob o cursor permanece no lugar. Nas visões de dia, semana útil e semana do Scheduler eles alteram a altura do slot.
- ScrollToTime(aTime) rola o eixo até um horário.

Na Timeline de recursos o período segue o zoom: um dia até pzHours, 4 semanas em pzDays, 3 meses em pzWeeks, um ano em pzMonths, 3 anos em pzQuarters e uma década em pzYears.

Exemplo

```
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;
```

Veja Também

- TsgcVCLGantt
- TsgcVCLScheduler
- Calendário de Trabalho
- Arrastar e Soltar

Arrastar e Soltar

Cada visão edita os dados com o mouse, uma caneta ou um dedo. Um arrasto começa depois que o ponteiro se move 4 DIPs, mostra uma prévia enquanto acontece, rola a visão ao chegar a uma borda e confirma a edição quando o botão é solto. Escape ou um segundo botão o cancela.

Os arrastos de cada visão

View	Arrastos
Scheduler	Arraste sobre um slot vazio para criar um evento (OnItemCreate dá o título ou o recusa). Mova um evento para outro horário, outro dia ou outra coluna de recurso, e redimensione-o na borda superior ou inferior. Movimentos e redimensionamentos se ajustam a Options.SlotMinutes. Na faixa de dia inteiro e na visão de mês, dias inteiros são criados e movidos. Na Timeline uma barra se move para outro horário ou outra linha de recurso e é redimensionada na escala.
Gantt	Mova uma barra, redimensione o seu início ou o seu término, arraste o progresso na metade inferior da barra, desenhe um vínculo de logo fora da extremidade de uma barra até outra barra (Registered) e arraste o divisor entre a árvore de tarefas e o gráfico. Um movimento mantém a duração de trabalho da tarefa.
Kanban	Mova um cartão para outra posição, outra coluna ou outra raia, ou reordene os cartões de uma coluna.

Permitindo e recusando

Configuração	Efeito
ReadOnly	Desativa todo arrastar e a tecla Delete do controle.
DragOptions	Os arrastos que o controle permite: pdoCreate, pdoMove, pdoResize, pdoProgress, pdoLink e pdoReorder. Padrão todos. No quadro, pdoReorder sem pdoMove mantém cada cartão na sua coluna.
OnItemChanging (dados)	Acionado uma vez por drop, em qualquer visão, com o item e os campos prestes a mudar. Defina aAccept como False para recusar a edição. É o lugar único para uma regra que deve valer em todas as visões.
OnCardMoving (Kanban)	Acionado antes de OnItemChanging com o status de destino e a raia.
OnItemCreate (Scheduler)	Acionado para um novo evento, com o seu início, término e recurso.
OnLinkCreating (Gantt)	Acionado para um novo vínculo. Pode alterar o tipo ou recusar o vínculo.
OnRecurrenceEdit (Scheduler)	Acionado para um drop sobre uma ocorrência de uma série. Veja Recorrência.
WIPMode pwmBlock (Kanban)	Impede que um cartão entre em uma coluna no seu limite de WIP. Veja Raias e Limites de WIP.

O que um drop faz

Um drop é um BeginUpdate e EndUpdate nos dados, de modo que cada visão se redesenha uma vez. Um movimento ou redimensionamento de início no Gantt também define a restrição da tarefa como pcSNET na nova data, a menos que a tarefa seja Manual, de modo que um agendamento posterior a mantém ali; com AutoSchedule o mecanismo roda após o drop. Quando o item tem um pai, o seu resumo é atualizado.

Teclado e toque

A tecla Delete aciona OnItemDelete para o item selecionado e o exclui quando aAccept continua True. No quadro, SelectNextCard move a seleção com as setas, Home e End. A entrada por toque chega como entrada de mouse, de modo que um dedo arrasta como um mouse.

Exemplo

```
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;
```

Veja Também

- Visões Sincronizadas
- TsgcVCLGantt

- TsgcVCLKanban
- TsgcVCLScheduler

Dependencies

Uma dependência liga dois itens: a sucessora (ToId) espera pela predecessora (FromId). Os vínculos são armazenados em Dependencies do TsgcPlannerData em todas as edições e salvos com JSON e XML do MS Project. O Gantt os desenha como setas, deixa o usuário desenhá-los com o mouse e os usa para agendar as tarefas na edição Registered.

Tipos e latência

Tipo	Significado
pdkFS (padrão)	Término a início: a sucessora começa quando a predecessora termina.
pdkSS	Início a início: a sucessora começa quando a predecessora começa.
pdkFF	Término a término: a sucessora termina quando a predecessora termina.
pdkSF	Início a término: a sucessora termina quando a predecessora começa.

Lag é em minutos de trabalho do Calendário de Trabalho, somados ao limite. Um lag negativo é uma antecipação, de modo que a sucessora se sobrepõe à predecessora. AddDependency(FromId, ToId, Kind, Lag) adiciona um vínculo em uma única chamada.

No Gantt

- ShowDependencies desenha cada vínculo como uma seta em cotovelo entre as barras visíveis. Os vínculos do caminho crítico usam a cor Critical.
- Arraste de logo fora do início ou do fim de uma barra e solte em outra barra para criar um vínculo. As extremidades escolhidas dão o tipo. OnLinkCreating pode alterar o tipo ou recusar o vínculo. Um vínculo para o próprio item, entre um item e o seu ancestral ou descendente, ou um que já existe é recusado.
- Um vínculo em uma tarefa de resumo vale para cada tarefa folha abaixo dela quando o plano é agendado.

Mantendo os vínculos consistentes

DeleteItem remove os vínculos do item excluído. Uma mudança de Id não renomeia os vínculos que a usam, portanto altere FromId e ToId desses vínculos na mesma atualização. Vínculos com FromId ou ToId desconhecido são mantidos, mas ignorados pelo mecanismo e não desenhados. Um ciclo de vínculos é relatado pelo mecanismo. Veja Agendamento Automático.
A origem DataSet lê vínculos de uma tabela própria (DependencyDataSet) ou de uma lista de predecessoras separadas por vírgula na tabela de tarefas (DependsOnField). Veja Vínculo com DataSet.

Exemplo

```
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;
```

Veja Também

- Agendamento Automático
- Caminho Crítico
- TsgcVCLGantt
- XML do MS Project

Agendamento Automático

Edição: Registered. Na edição Community, Schedule e SetBaseline do Gantt geram EsgcPlannerError, AutoSchedule não tem efeito e a unit do mecanismo não está incluída. Veja Recursos por Edição.
O mecanismo de agendamento, TsgcPlannerScheduleEngine de sgcPlanner_AutoSchedule, coloca as tarefas o mais cedo possível no calendário de trabalho após as suas predecessoras, calcula a folga de cada tarefa e marca o caminho crítico. O Gantt o executa com Schedule, ou após cada edição quando AutoSchedule é True, e o seu código pode executá-lo em um TsgcPlannerData sem nenhuma visão, por exemplo em um serviço.

As regras

- Uma tarefa folha mantém a sua duração de trabalho, os minutos de trabalho entre o seu Start e o Finish, e começa no primeiro minuto de trabalho igual ou posterior ao maior entre o início do projeto, a sua data de restrição e os limites das suas dependências.
- O início do projeto é o ProjectStart dos dados, ou o Start mais antigo das tarefas folha quando ProjectStart não está definido.
- A restrição pcSNET com um ConstraintDate significa início não antes dessa data. pcASAP, o padrão, não tem data. Mover uma barra no Gantt define pcSNET na nova data, de modo que o próximo agendamento mantém a tarefa onde foi colocada.
- Uma tarefa Manual, um item recorrente e toda tarefa de um ciclo de dependência mantêm as suas datas e continuam conduzindo as suas sucessoras.
- Uma tarefa sem minutos de trabalho é um marco e fica exatamente no seu limite.
- Uma tarefa de resumo assume o início mais antigo e o término mais recente das suas folhas. Um vínculo em um resumo vale para cada uma das suas folhas.

Executando

Membro	Descrição
Gantt.Schedule	Executa o mecanismo nos dados do Gantt. Retorna False quando um ciclo foi encontrado.
Gantt.AutoSchedule	Executa o mecanismo após cada edição feita no Gantt: um movimento, um redimensionamento e um novo vínculo. Alterações feitas no código ou em outra visão não o executam, portanto chame Schedule depois delas.
Gantt.OnScheduleError, Engine.OnScheduleError	Acionado após a atualização quando os vínculos formam um ciclo, com a mensagem e os Ids das tarefas do ciclo. As outras tarefas são agendadas mesmo assim.
Engine.Schedule(aData), Engine.CycleIds	A mesma passagem sem uma visão, e os Ids do último ciclo encontrado.

Todas as datas e resultados são gravados em uma única atualização: pckItems para as datas e pckSchedule para a folga e os indicadores críticos. Uma passagem que não altera nada não notifica nada. Em uma máquina de desenvolvimento, 10.000 tarefas com 15.000 vínculos e 30 feriados são agendadas em cerca de 11 ms (Win64).

Exemplo

```
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;
```

Veja Também

- Dependências
- Caminho Crítico
- Linhas de Base
- Calendário de Trabalho

Caminho Crítico

Edição: Registered. Na edição Community, ShowCriticalPath não tem efeito e o mecanismo que calcula a folga não está incluído. Veja Recursos por Edição.
O caminho crítico é a cadeia de tarefas que define o fim do projeto: um atraso em qualquer uma delas atrasa o término. O mecanismo de agendamento o encontra a cada passagem.

Folga

- Depois da passagem para a frente, uma passagem para trás a partir do término do projeto, o término mais tardio de todas as tarefas folha, dá **TotalSlack**: os minutos de trabalho que uma tarefa pode atrasar sem mover o término do projeto. É o menor entre a folga de início e a folga de término.
- **Critical** é True quando TotalSlack é 0 ou menos. A folga é negativa quando uma tarefa que mantém as suas datas (Manual ou recorrente) quebra uma dependência.
- Uma tarefa de resumo assume a menor folga das suas folhas e é crítica quando uma delas é. As tarefas de um ciclo não recebem folga e nunca são críticas.

TotalSlack e Critical são resultados somente leitura. Não são salvos no JSON, portanto execute Schedule depois de carregar um projeto.

Mostrando

Defina ShowCriticalPath do Gantt como True: as barras e os vínculos das tarefas críticas são desenhados com a cor Critical do tema, que Colors.Critical substitui. Execute Schedule, ou defina AutoSchedule, para que os resultados estejam atualizados.

Exemplo

```
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;
```

Veja Também

- Agendamento Automático
- Dependências
- Linhas de Base
- TsgcVCLGantt

Linhas de Base

Edição: Registered. Na edição Community, BaselineStart e BaselineFinish são armazenados e salvos, ShowBaseline não tem efeito e SetBaseline do Gantt gera EsgcPlannerError. Veja Recursos por Edição.
Uma linha de base é uma cópia salva do plano. Cada item tem BaselineStart e BaselineFinish ao lado do seu Start e Finish, e o Gantt desenha a linha de base como uma barra fina sob a barra atual, de modo que uma tarefa que atrasou é vista de relance.

Definindo a linha de base

- SetBaseline do Gantt copia Start e Finish de cada item para as suas datas de linha de base, em uma única atualização.
- Sem uma visão, SetBaseline e ClearBaseline de TsgcPlannerScheduleEngine fazem o mesmo em um TsgcPlannerData. ClearBaseline volta as datas de linha de base para 0, que significa não definido.
- As datas de linha de base são propriedades comuns, de modo que o seu código pode definir a linha de base de uma tarefa, ou manter várias linhas de base próprias e copiar uma delas.

Mostrando

Defina ShowBaseline do Gantt como True. As linhas dos itens com linha de base desenham a barra atual na parte superior da linha e a linha de base abaixo dela. Compare as datas no código com WorkingMinutesBetween do calendário para relatar o atraso em tempo de trabalho.

Arquivos

O JSON salva as datas de linha de base. O XML do MS Project lê e grava a linha de base número 0, e o iCalendar as mantém em X-SGC-BASELINE-START e X-SGC-BASELINE-FINISH.

Exemplo

```
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;
```

Veja Também

- Agendamento Automático
- Caminho Crítico
- TsgcVCLGantt
- XML do MS Project

Raias e Limites de WIP

Edição: Registered. Na edição Community, SwimlaneBy age como pslNone e WIPLimit não mostra contador e nunca bloqueia um cartão. Colunas, larguras e recolhimento funcionam em todas as edições. Veja Recursos por Edição.
O quadro Kanban mostra os itens como cartões em colunas. As raias dividem o quadro em faixas horizontais, e os limites de WIP limitam o trabalho em andamento de uma coluna.

Columns

Propriedade	Descrição
ColumnSource	pkcStatuses (padrão): uma coluna por status dos dados, na ordem deles. pkcList: apenas as colunas de Columns.
Columns	Entradas com StatusId, Caption, Color, Width, WIPLimit e Collapsed. Com pkcStatuses uma entrada com o StatusId de um status acrescenta as suas configurações a essa coluna. AddColumn(StatusId, Caption, WIPLimit) adiciona uma.
ShowUnknownColumn	Mostra os cartões cujo status não corresponde a nenhuma coluna em uma última coluna. Não é possível soltar um cartão nela.

Um item de resumo e uma ocorrência modificada não são mostrados como cartões. Os cartões de uma coluna são ordenados por Rank, depois pela sua ordem nos dados. CollapseColumn e ExpandColumn, ou o chevron do cabeçalho da coluna, recolhem uma coluna em um trilho estreito, e OnColumnCollapse o relata.

Raias

SwimlaneBy	Raias	Um drop em outra raia
pslNone (padrão)	Sem raias.	Não é possível.
pslResource	Uma raia por recurso. Um cartão vai para a raia do seu primeiro recurso.	Substitui o primeiro recurso do cartão.
pslParent	Uma raia por pai de um cartão, na ordem dos dados.	Define ParentId do cartão.
pslTag	Uma raia por primeira tag de um cartão, ordenadas.	Não é possível.
pslEvent	As raias que OnGetSwimlane retorna como uma chave e um título, em ordem de aparição. Sem um handler age como pslNone.	Não é possível.

Cartões sem raia vão para uma raia no final.

Limites de WIP

- WIPLimit de uma entrada de coluna é o maior número de cartões que a coluna deve conter, contados em todas as raias. 0 significa sem limite.
- O contador do cabeçalho mostra a contagem, a contagem e o limite abaixo do limite, um contador âmbar no limite e um vermelho acima dele.

- WIPMode pwmWarn (padrão) apenas marca a coluna. pwmBlock recusa um cartão que entraria em uma coluna no seu limite, enquanto os cartões que já estão na coluna ainda podem ser reordenados.

Exemplo

```
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;
```

Veja Também

- TsgcVCLKanban
- Arrastar e Soltar
- Temas
- TsgcFMXKanban

Vínculo com DataSet

TsgcPlannerSource_DataSet vincula um TsgcPlannerData a TDataSets: qualquer dataset da unit DB, como FireDAC, TClientDataSet, dbExpress ou um dataset de terceiros. Ele carrega os itens, os recursos e as dependências, acompanha as alterações dos datasets e grava de volta as edições das visões. Usa apenas a unit DB, de modo que funciona em VCL, FireMonkey e aplicativos de console, em todas as edições.

Datasets e mapas de campos

Dataset	Mapa de campos	Nomes de campo padrão
DataSet	ItemFields	Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, RecurrenceRule e DependsOn.
ResourceDataSet	ResourceFields	Id, Name, Color e Initials.
DependencyDataSet	DependencyFields	FromId, ToId, Kind e Lag.

Um nome de campo vazio ou não encontrado no dataset é ignorado, de modo que uma tabela mapeia apenas os campos que tem. As datas são lidas de campos de data e hora, de campos numéricos ou de texto ISO 8601; os booleanos de campos booleanos, números ou texto como Y ou True; as cores de números ARGB ou texto como #RRGGBB. Um Finish nulo dá Finish = Start, ou a próxima meia-noite para um item de dia inteiro. O tipo (Kind) de uma dependência é FS, SS, FF ou SF, e o seu Lag é em minutos de trabalho. Sem um DependencyDataSet, o campo DependsOn pode guardar os Ids separados por vírgula das predecessoras de uma tarefa, lidos como vínculos término a início. OnLoadItem é acionado para cada registro depois que os campos mapeados são copiados, para preencher outras propriedades do item.

Carregamento e atualizações ao vivo

Active True carrega todo dataset aberto em uma única atualização, de modo que as visões vinculadas se redesenham uma vez; 10.000 linhas carregam em cerca de 26 ms no Win64. Com LiveUpdates, uma edição ou inserção postada atualiza ou adiciona somente esse item, e uma mudança no número de registros (uma exclusão, um filtro ou um refresh) sincroniza os itens por Id. Rolar e edições não postadas não alteram nada. Load e Refresh recarregam manualmente.

Gravação de volta

Com WriteBack, True por padrão, uma edição dos dados, feita em uma visão ou no código, é postada no registro encontrado com Locate no campo Id: apenas os campos mapeados alterados são atribuídos. Um novo item acrescenta um registro, um item excluído exclui o seu registro, e vínculos novos ou removidos são gravados no DependencyDataSet ou na lista DependsOn. Enquanto o dataset está em estado de edição ou inserção nada é gravado, e as alterações pendentes são gravadas quando ele volta ao estado de navegação. Recursos, campos calculados, campos de lookup e campos somente leitura nunca são gravados. Uma gravação nunca volta como um novo carregamento.

Exemplo

```
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;
```

A demo DataSetPlanner vincula três tabelas TClientDataSet com os seus próprios nomes de campo. Veja Demos.

Veja Também

- TsgcPlannerSource_DataSet
- Dados Compartilhados
- Persistência JSON
- Dependências

Persistência JSON

Um TsgcPlannerData salva em JSON e carrega a partir dele em todas as edições e em todas as plataformas: SaveToJSON e LoadFromJSON para uma string, SaveToFile e LoadFromFile para um arquivo, e as funções de sgcPlanner_JSON para streams (sgcPlannerJSONSaveToStream e sgcPlannerJSONLoadFromStream) e arquivos.

O formato

O documento traz `"format": "sgcPlanner"` e uma versão, depois o início do projeto, o calendário, os recursos, os status, os itens e as dependências. As datas são horários locais ISO 8601 e as cores são números ARGB. Um projeto reduzido:

```
{
  "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"}
  ]
}
```

Regras

- Arquivos e streams são UTF-8 sem marca de ordem de bytes. Uma marca de ordem de bytes é aceita ao carregar.
- Um membro de item que guarda o seu valor padrão (um texto vazio, false, 0 ou uma data não definida) não é gravado, exceto id, title, start e finish, o que mantém os arquivos pequenos.
- Os valores calculados (TotalSlack, Critical, IsSummary), a seleção e Version não são salvos.
- Carregar substitui todos os dados, inclusive o calendário e o ProjectStart, em uma única atualização, de modo que cada visão vinculada se redesenha uma vez. Um membro ausente carrega o seu valor padrão.
- Membros desconhecidos, membros de outro tipo JSON e valores que não são interpretados são ignorados, de modo que um arquivo gravado por uma versão posterior ainda carrega. Uma versão ausente é lida como 1.
- Quando o texto não é JSON válido, a raiz não é um objeto, o formato é outro ou a versão não é suportada, EsgcPlannerError é gerado e os dados ficam como estavam.
- Antes do Delphi 2009 o texto passa pela página de código ANSI do sistema.

Em uma máquina de desenvolvimento, 10.000 itens com 15.000 vínculos são salvos em cerca de 48 ms e carregados em cerca de 56 ms (Win64).

Exemplo

```
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;
```

Veja Também

- TsgcPlannerData
- Vínculo com DataSet
- iCalendar
- XML do MS Project

iCalendar

Edição: Registered. Na edição Community, TsgcPlannerICalendar não existe; JSON é o formato de arquivo dessa edição. Veja Recursos por Edição.
TsgcPlannerICalendar importa e exporta calendários RFC 5545 (.ics), o formato do Google Agenda, do Outlook, do Apple Calendar e da maioria dos servidores de calendário. Um VEVENT é um item.

Importação

- ImportFromString, ImportFromStream e ImportFromFile leem UTF-8, com ou sem marca de ordem de bytes. UTF-8 inválido é lido como Latin-1, com um aviso.
- ImportMode pimReplace (padrão) exclui primeiro os itens e as dependências e mantém os recursos, os status, o calendário e o ProjectStart. pimMerge atualiza o item com o mesmo Id e adiciona os outros.
- Horários em UTC e horários com um TZID são convertidos para o horário local da máquina, com o VTIMEZONE do arquivo. UTC, GMT e Etc/UTC não precisam de VTIMEZONE. Um TZID desconhecido gera um aviso e é lido como horário local.
- O Id do item é o UID. Um VEVENT com um RECURRENCE-ID torna-se uma ocorrência modificada da série com esse UID.
- Problemas que não interrompem a importação, como uma parte de RRULE não suportada ou um RDATE, são relatados por OnImportWarning com o número da linha e contados em WarningCount. Um texto sem BEGIN:VCALENDAR gera EsgcPlannerError e deixa os dados inalterados.

Exportação

- ExportToString, ExportToStream e ExportToFile gravam cada item como um VEVENT, com ProductId como PRODID.
- Com ExportUTC (padrão True) os horários dos itens que não se repetem são gravados em UTC. Séries e ocorrências modificadas sempre usam horários locais flutuantes, de modo que as suas ocorrências ficam na mesma hora nas mudanças de horário de verão.
- Um item de dia inteiro cujo início e término são meia-noite é gravado como valores DATE.
- As dependências não são exportadas.

Mapeamento

iCalendar	Item
UID, SUMMARY, DESCRIPTION	Id, Title, Notes.
DTSTART, DTEND	Start, Finish.
RRULE, EXDATE, RECURRENCE-ID	RecurrenceRule, RecurrenceExceptions, RecurrenceParentId e RecurrenceDate.
STATUS	Status em minúsculas: tentative, confirmed ou cancelled. Outro status é mantido em X-SGC-STATUS.
RESOURCES	Os nomes dos recursos. X-SGC-RESOURCES guarda os seus Ids quando diferem, e a importação cria os recursos ausentes.
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	As propriedades do planner para as quais o iCalendar não tem campo, de modo que um arquivo do planner faz a viagem de ida e volta.

Em uma máquina de desenvolvimento, 5.000 VEVENTs com fusos horários são importados em cerca de 84 ms (Win64). A demo ICalendarSync importa calendários do Google e do Outlook e exporta uma série. Veja Demos.

Veja Também

- TsgcPlannerICalendar
- Recorrência
- XML do MS Project
- Persistência JSON

XML do MS Project

Edição: Registered. Na edição Community, TsgcPlannerMSProject não existe; JSON é o formato de arquivo dessa edição. Veja Recursos por Edição.
TsgcPlannerMSProject importa e exporta o XML do Microsoft Project (MSPDI), o formato XML que o Microsoft Project abre e salva. Ele lê tarefas, vínculos, o calendário, recursos, atribuições e linhas de base.

Importação

- ImportFromString, ImportFromStream e ImportFromFile leem UTF-8, com ou sem marca de ordem de bytes. Um documento que não está bem formado gera EsgcPlannerError com o número da linha.
- ImportMode pmiReplace (padrão) exclui os itens e os vínculos e toma o calendário e o início do projeto (StartDate) do arquivo. Os recursos e os status são mantidos, e os recursos do arquivo são adicionados ou renomeados.
- pmiMerge mantém o calendário e o ProjectStart, atualiza os itens por Id e adiciona ou atualiza os vínculos pelas suas duas extremidades. Os recursos dos itens seguem o arquivo somente quando ele tem Assignments.
- OnImportWarning relata o que é lido de outra maneira, por exemplo um tipo de restrição que o sgcPlanner não tem.

Mapeamento

MSPDI	sgcPlanner
Task UID, Name, Start, Finish, PercentComplete, Milestone, Notes	Id, Title, Start, Finish, Progress, Milestone, Notes.
OutlineLevel e a ordem das tarefas	ParentId, a árvore de tarefas.
PredecessorLink Type 0, 1, 2, 3	pdkFF, pdkFS, pdkSF, pdkSS. LinkLag é em décimos de minuto de trabalho; os formatos de latência decorrida e percentual são convertidos.
ConstraintType 0 e 4	pcASAP e pcSNET com ConstraintDate. O tipo 2 (deve começar em) é lido como pcSNET e os outros tipos como pcASAP, ambos com um aviso.
Baseline Number 0	BaselineStart e BaselineFinish.
Calendar WeekDays, WorkingTimes, Exceptions	O calendário de trabalho: dias úteis, horário de trabalho e feriados.
Resource Code ou Name, Assignments	Os recursos e a lista Resources de cada item.

Exportação

ExportToString, ExportToStream e ExportToFile gravam a ordem do esquema do SaveVersion 14: um calendário Standard com uma exceção por feriado, as tarefas na ordem da estrutura de tópicos, os seus vínculos e as suas atribuições. Um Id numérico mantém o seu UID e os outros Ids recebem o próximo número livre. O MSPDI mantém um vínculo por par de tarefas. Status, tags, cor, rank, dia inteiro e recorrência não são gravados, portanto mantenha um arquivo JSON ao lado quando precisar deles.
Em uma máquina de desenvolvimento, 10.000 tarefas com 15.000 vínculos são importadas em cerca de 264 ms (Win64). A demo ProjectGantt abre e salva arquivos MSPDI. Veja Demos.

Veja Também

- TsgcPlannerMSProject
- Dependências
- Linhas de Base
- Persistência JSON

Renderização sem Janela

Toda visão do sgcPlanner pode ser desenhada sem janela. Os modelos de visão desenham em um TsgcUICanvas do sgcUICore, e os mesmos desenhadores servem aos controles em um formulário, ao TsgcPlannerRenderer e aos métodos SaveToPNG e SaveToSVG dos controles, de modo que uma imagem tem os mesmos pixels da tela.

TsgcPlannerRenderer

O renderer é um componente não visual sem unit VCL ou FireMonkey, de modo que roda em um serviço, em um servidor web, em um aplicativo de console e no Linux64. Funciona em todas as edições.

Membro	Descrição
Data, View	Os dados e a visão a desenhar: prvScheduler, prvGantt ou prvKanban.
Width, Height, Scale	O tamanho em DIPs (padrão 800 x 600) e a escala da saída, 2 para uma imagem HiDPI.
Theme, Colors	As mesmas substituições de tema e de cores dos controles. Veja Temas.
SchedulerOptions, GanttOptions, KanbanOptions	As opções publicadas de cada visão. Assign também aceita um modelo de visão, de modo que `Renderer.GanttOptions.Assign(VCLGantt1.GanttView)` copia o que um formulário mostra.
Date	A data que o Scheduler mostra, 0 para hoje.
SchedulerView, GanttView, KanbanView	Os modelos de visão que o renderer possui, para ExpandAll, ZoomToFit, ScrollToTime ou CollapseColumn antes de uma renderização.
RenderTo(aCanvas), SaveToPNG, SaveToSVG	Desenha em um canvas, ou salva um PNG ou um SVG em um arquivo ou em um stream.
FontFile, FontFileBold	Arquivos TrueType para o texto. Vazio usa a fonte embutida no sgcUICore, que não precisa de nenhuma fonte instalada na máquina.

Imagens dos controles

Os seis controles também têm SaveToPNG e SaveToSVG, com um nome de arquivo ou um stream e largura, altura e escala opcionais. O tamanho é em DIPs e 0 é o tamanho do cliente; uma escala 0 é a escala do controle. A saída mantém a posição de rolagem do controle e é a mesma imagem de um renderer com os mesmos dados, opções, tamanho e escala. As barras de rolagem, a prévia do arrasto e as marcas de drop ficam de fora.

Em um servidor

- A seleção da imagem é SelectedId dos dados; limpe-a para um relatório sem item selecionado.
- Os nomes de dias e meses e os formatos de data vêm de sgcPlannerLocale, que é inicializado a partir da máquina. Defina-o na inicialização quando um servidor precisar produzir os mesmos rótulos de um desktop.
- A saída SVG mantém o texto como texto, de modo que fica nítido em qualquer zoom e pode ser pesquisado.

Exemplo

```
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;
```

A demo HeadlessReport é um aplicativo de console para Windows e Linux64 que grava cada visão como PNG e SVG. Veja Demos.

Veja Também

- TsgcPlannerRenderer
- Temas
- Plataformas
- Visões Sincronizadas

Temas

Os controles do planner e o renderer são desenhados com os temas do sgcUICore, de modo que um planner combina com os outros produtos sgcUI no mesmo formulário, na VCL, no FireMonkey e na saída sem janela.

Tema e cores

- **Theme** é utLight (padrão), utDark ou utCustom. utCustom parte das cores claras.
- **Colors** substitui cores individuais do tema: Background, Surface, Border, Grid, HeaderBackground, HeaderText, Text, TextMuted, Accent, Selection, NonWorking, NowLine, Critical e Ghost. Uma cor é um TsgcUIColor ARGB, por exemplo sgcUIARGB($FF, $00, $96, $88). 0 mantém a cor do tema.

Cores dos itens

Um item é desenhado com a primeira cor definida: a Color do item, a Color do seu status, a Color do seu primeiro recurso e, caso contrário, uma cor da paleta do tema escolhida a partir do seu Id, de modo que o mesmo item mantém a mesma cor em todas as visões e em todas as execuções.

Configuração	Efeito
BarColorMode do Gantt	pbcItem (a cor do item acima), pbcStatus (a cor do status) ou pbcResource (a cor do primeiro recurso).
Color das colunas do Kanban	A barra de cor do cabeçalho de uma coluna. Com pkcStatuses a cor do status é usada quando não está definida.
Tags	As pílulas de tag dos cartões assumem uma cor da paleta do tema escolhida a partir do texto da tag.
Resources	Os círculos de iniciais dos cartões e das raias usam a cor do recurso.

Fontes e tamanhos

O texto usa a fonte embutida no sgcUICore, de modo que a saída é a mesma em qualquer máquina. O renderer aceita arquivos TrueType com FontFile e FontFileBold. Os tamanhos são em DIPs e são escalados para o monitor, o formulário ou a Scale do renderer.

Exemplo

```
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;
```

Veja Também

- Renderização sem Janela
- Raias e Limites de WIP
- TsgcVCLGantt
- TsgcVCLKanban

Recursos por Edição

sgcPlanner é distribuído nas edições Registered, Trial e Community. A Trial tem todos os recursos da Registered. Esta tabela lista cada recurso e a edição de que ele precisa. Veja Edições para as edições em si.

Recurso	Registered e Trial	Community
Dados Compartilhados, Visões Sincronizadas	Sim	Sim
Calendário de Trabalho	Sim	Sim
Recorrência	Sim	Não. Um item recorrente é mostrado uma vez.
Escala de Tempo e Zoom	Sim	Sim
Arrastar e Soltar	Sim	Sim, exceto arrastar para vincular no Gantt.
Dependências	Sim	Os vínculos são armazenados e salvos, mas não desenhados.
Agendamento Automático	Sim	Não. Schedule gera EsgcPlannerError.
Caminho Crítico	Sim	No.
Linhas de Base	Sim	As datas são armazenadas e salvas, mas não desenhadas. SetBaseline gera EsgcPlannerError.
Scheduler: visões de dia, semana útil, semana, mês e agenda	Sim	Sim
Timeline de recursos do Scheduler	Sim	Não. A visão mostra um aviso.
Colunas do Kanban, recolhimento e opções de cartão	Sim	Sim
Raias e Limites de WIP	Sim	Não. SwimlaneBy age como pslNone.
Vínculo com DataSet	Sim	Sim
Persistência JSON	Sim	Sim
iCalendar, XML do MS Project	Sim	Não. Os componentes não existem.
Renderização sem Janela, Temas	Sim	Sim

Código que compila em todas as edições

Os recursos avançados são compilados sob SGC_PLANNER_ADVANCED, que o sgcVer.inc define nas edições Registered e Trial e desativa na Community. Inclua sgcVer.inc em uma unit que precise compilar nas duas, e coloque o código dos recursos avançados dentro de `{$IFDEF SGC_PLANNER_ADVANCED}`. As propriedades das visões são publicadas em todas as edições, de modo que um formulário que define ShowCriticalPath ou SwimlaneBy carrega na Community; um formulário que contém TsgcPlannerICalendar ou TsgcPlannerMSProject não carrega.

Exemplo

```
{$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;
```

Veja Também

- Edições
- Instalação
- Solução de Problemas

Sincronização com o Google Agenda

TsgcPlannerGoogleSync mantém um TsgcPlannerData e um Google Agenda em sincronização bidirecional. Este tópico descreve o que o componente precisa, como uma sincronização é executada, como os campos são mapeados, a regra de conflito e o arquivo de estado.

Requisitos

- A edição Registered do sgcPlanner, com o pacote de ponte sgcPlannerWS (as units sgcPlannerWS_Classes, sgcPlannerWS_Google e sgcPlannerWS_ICal).
- sgcWebSockets, edição Standard ou superior, que inclui o cliente do Google Agenda TsgcHTTPGoogleCloud_Calendar_Client. Coloque os fontes do sgcPlanner, do sgcUICore e do sgcWebSockets, ou a pasta da biblioteca do sgcWebSockets, no caminho de pesquisa.
- As bibliotecas OpenSSL que o sgcWebSockets usa para HTTPS, ao lado do executável.

O projeto do Google Cloud e o cliente OAuth2

1. Abra o console do Google Cloud, crie um projeto (ou escolha um) e ative a Google Calendar API.
2. Configure a tela de consentimento OAuth. Enquanto o aplicativo estiver em teste, adicione como usuários de teste as contas do Google que farão login.
3. Crie um ID de cliente OAuth do tipo Aplicativo para computador e copie o ID do cliente e o segredo do cliente.
4. Configure o cliente: GoogleCloudOptions.Authentication como gcaOAuth2, o ID e o segredo do cliente em GoogleCloudOptions.OAuth2, LocalIP e LocalPort para o redirecionamento do navegador (por exemplo 127.0.0.1 e 8080), e Scopes.EventsReadWrite e Scopes.CalendarsReadWrite. A sincronização lê o calendário uma vez para obter seu fuso horário.
5. Na primeira requisição, o navegador abre a página de consentimento do Google. Quando o usuário permite o acesso, a sincronização continua.

Não guarde o segredo do cliente em um arquivo de formulário nem em código-fonte que você distribui. A demo o digita em tempo de execução e o mantém apenas na memória.

Como uma sincronização é executada

1. Thread de trabalho: a lista de eventos de CalendarId, com o token de sincronização da última sincronização (apenas as alterações desde então, incluindo eventos excluídos), ou todos os eventos na primeira sincronização, seguindo nextPageToken. Cada série chega como um único evento mestre.
2. Thread principal: os eventos são aplicados a Data dentro de um único BeginUpdate e EndUpdate. Um evento cancelado exclui seu item, uma ocorrência cancelada adiciona uma exceção à sua série. Os conflitos são resolvidos e a lista de envio é montada com os itens novos, alterados e excluídos, depois de OnFilterItem.
3. Thread de trabalho: inserções, atualizações e exclusões no Google.
4. Thread principal: os novos ids de evento, etags, horários de atualização, impressões digitais e o novo token de sincronização são armazenados, o estado é salvo e OnSyncDone é disparado.

Um item lido do Google recebe o id `g:` mais o id do evento. Um item criado no planner mantém seu id, que viaja nas propriedades estendidas privadas do seu evento (sgcPlannerId), de modo que volta com o mesmo id.

Mapeamento de campos

Item do planner	Evento do Google	Observações
Title	summary	

Observações	description	
Start, Finish	start.dateTime, end.dateTime	Lidos com seu deslocamento e convertidos para a hora local. Gravados em UTC, exceto os eventos recorrentes e suas instâncias, que são gravados como hora local com seu deslocamento mais timeZone (TimeZone, ou o fuso horário do calendário). Os horários têm precisão de minuto, os segundos são descartados.
AllDay, Start, Finish	start.date, end.date	O fim é exclusivo nos dois lados: um item de dia inteiro em 1 de outubro vai de 1 de outubro 00:00 a 2 de outubro 00:00, e seu evento termina em 2 de outubro. Ao gravar, um Finish que não está à meia-noite é arredondado para o dia seguinte.
RecurrenceRule	the RRULE line of recurrence	Repassada sem alteração, nos dois sentidos.
RecurrenceExceptions	the EXDATE lines of recurrence	Gravadas como horários UTC, ou como datas nas séries de dia inteiro, de modo que não é necessário um banco de dados de fusos horários.
RecurrenceParentId, RecurrenceDate	recurringEventId, originalStartTime	Uma ocorrência modificada de uma série.
Id	extendedProperties.private.sgcPlannerId	O id do item no planner.

Outras linhas de recorrência de um evento, como RDATE ou EXRULE, são mantidas no estado e gravadas de volta sem alteração. Uma regra que o mecanismo de recorrência do sgcPlanner não expande (por exemplo uma com BYSETPOS ou HOURLY) também é mantida e gravada de volta sem alteração; o item mostra sua primeira ocorrência e OnError informa o motivo uma vez por regra, de modo que o Google nunca perde uma regra.

O que não é sincronizado

Status, Progress, Color, Tags, Resources e as dependências dos itens não são enviados ao Google, e os participantes, lembretes, cores e dados de conferência dos eventos não são lidos. Um item editado no planner é gravado com o método patch da Calendar API, que envia apenas os campos de que o planner é dono, de modo que os participantes, lembretes, dados de conferência e outras propriedades do evento são mantidos.

Conflitos

Um item alterado no planner e no Google desde a última sincronização é um conflito. O componente registra quando um item passou a ser diferente da sua última impressão digital sincronizada; esse momento é comparado com o horário `updated` do evento, em UTC:

- A alteração mais recente vence. Um empate, ou um evento sem horário de atualização legível, fica com o Google.
- OnConflict recebe o item e o JSON do evento e pode definir aResolution como gcrKeepLocal, gcrKeepRemote ou gcrSkip (os dois lados permanecem como estão nesta execução).
- Uma edição contra uma exclusão mantém a edição nos dois lados: um evento excluído é criado de novo, ou um item excluído é restaurado. Conta como conflito, mas não dispara OnConflict.
- Um evento cujo conteúdo já coincide com o item é apenas registrado, não contado como conflito. Os eventos que voltam do próprio envio do componente são reconhecidos pelo etag.
- Com Direction gsdPullOnly as alterações locais nunca contam, portanto o Google sempre vence.

Este manipulador registra os dois títulos, mantém a versão local dos itens com a tag `office` e mantém fora do Google os itens com a tag `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;
```

O arquivo de estado

O estado é JSON (UTF-8): o nome e a versão do formato, o id do calendário, o token de sincronização, o fuso horário e, para cada item sincronizado, o id local, o id do evento, o etag, o horário `updated`, a impressão digital dos campos sincronizados, o momento em que o item foi alterado e as linhas de recorrência extras. StateFileName o carrega antes da primeira sincronização e o salva após cada uma, por meio de um arquivo temporário que depois é renomeado. SaveStateToStream e LoadStateFromStream o guardam em outro lugar. O estado de outro calendário é apagado quando CalendarId muda.
Mantenha o estado junto com os itens que ele descreve. Um item sincronizado que falta em Data conta como excluído localmente, e a próxima sincronização exclui seu evento no Google. Quando os itens não são salvos, deixe StateFileName vazio, de modo que cada inicialização comece com uma sincronização completa em um planner vazio. ResetState força uma sincronização completa sem excluir nada.

Erros e sincronizações completas

- Um erro ao listar não altera nada. Um erro ao enviar para na primeira chamada que falha. O estado mantém o que teve êxito e o token de sincronização antigo, de modo que a próxima sincronização lê de novo as mesmas alterações remotas e tenta o restante outra vez. OnError informa o motivo.
- Quando o Google responde 410 Gone a um token de sincronização expirado, o token é apagado e a execução se repete como uma lista completa. O mapeamento dos itens é mantido, de modo que nada é inserido duas vezes. Um evento mapeado que falta na lista completa conta como excluído no Google, com a regra de que uma edição vence uma exclusão.
- Uma atualização respondida com 404 ou 410 é enviada de novo como inserção. Uma exclusão respondida com 404 ou 410 conta como concluída.

Sincronização com o Google e assinatura iCalendar juntas

Um TsgcPlannerICalSubscription pode mesclar um feed somente leitura nos mesmos dados. Seus itens recebem o IdPrefix `ics:`, que é o ExcludeIdPrefix padrão da sincronização com o Google, de modo que os eventos do feed são exibidos, mas nunca enviados ao Google. Quando você mudar um prefixo, mude o outro.

Demos

As demos Delphi e C++Builder GoogleCalendarSync e ICalSubscription mostram os dois componentes em um TsgcVCLScheduler. Execute-as com `--test` para um autoteste offline com dados de teste, sem conta do Google. Consulte Demos.

Veja também

- TsgcPlannerGoogleSync
- TsgcPlannerICalSubscription
- Sincronização com o Google Agenda e iCalendar no .NET
- Recorrência
- iCalendar
- TsgcPlannerData

TsgcPlannerGoogleSync

Unit: sgcPlannerWS_Google
TsgcPlannerGoogleSync mantém os itens de um TsgcPlannerData em sincronização bidirecional com um Google Agenda. Faz parte do pacote de ponte sgcPlannerWS (edição Registered) e chama a Google Calendar API por meio do **TsgcHTTPGoogleCloud_Calendar_Client** do sgcWebSockets. O cliente é usado apenas como transporte autenticado: o componente chama seus métodos de nível de API (Event_List, Event_Insert, Event_Patch, Event_Delete e Calendar_Get) e lê e grava o JSON dos eventos por conta própria, de modo que o id de calendário `primary` funciona e o OnError do cliente fica para a aplicação. Consulte Sincronização com o Google Agenda para a configuração do Google Cloud, o mapeamento de campos, a regra de conflito e o arquivo de estado.

- Uma sincronização lista os eventos alterados (com o token de sincronização armazenado, ou todos se não houver, seguindo nextPageToken), aplica-os a Data no thread principal, depois envia as alterações locais (inserções, atualizações e exclusões) e armazena no estado os novos ids de evento, etags, impressões digitais e o token de sincronização.
- As alterações locais são encontradas pela impressão digital: um item cujos campos sincronizados diferem da última sincronização é enviado. Não é preciso um horário de modificação por item.
- Um item alterado nos dois lados é um conflito. A alteração mais recente vence, um empate fica com o Google e OnConflict pode substituir isso. Uma edição sempre vence uma exclusão.
- Um erro de HTTP ou de autorização interrompe a execução, mantém o token de sincronização antigo (de modo que a próxima sincronização tenta de novo) e dispara OnError.

Propriedades

Propriedade	Tipo	Descrição
Active	Boolean	Padrão True. Sync e SyncAsync não fazem nada e retornam False enquanto Active é False. A sincronização automática de AutoSyncInterval só é executada enquanto Active estiver ativo.
AutoSyncInterval	Integer	Segundos entre duas sincronizações automáticas, 0 (padrão) as desativa. Enquanto Active estiver ativo, um valor acima de 0 sincroniza imediatamente em um thread de trabalho e depois a cada AutoSyncInterval segundos.
CalendarId	string	O calendário a sincronizar. Padrão `primary`, o calendário principal da conta. O id de outro calendário (mostrado nas configurações dele no Google Agenda) também funciona.
Client	TsgcHTTPGoogleCloud_Calendar_Client	O cliente do Google Agenda do sgcWebSockets que autentica e envia as requisições. A aplicação não deve usá-lo enquanto Busy estiver ativo.
Data	TsgcPlannerData	Os dados do planner a sincronizar.
Direction	TsgcPlannerGoogleDirection	gsdTwoWay (padrão) lê e grava. gsdPullOnly nunca grava no Google e as alterações locais nunca contam, portanto o Google sempre vence. gsdPushOnly apenas envia as alterações locais e não lê as alterações feitas no Google.
ExcludeIdPrefix	string	Os itens cujo Id começa com ele nunca são enviados. Padrão `ics:`, o IdPrefix padrão de TsgcPlannerICalSubscription, de modo que os eventos de uma assinatura não vão para o Google.

StateFile-Name	string	O arquivo JSON do estado da sincronização. É carregado antes da primeira sincronização e salvo após cada uma. Vazio (padrão) mantém o estado apenas na memória.
TimeZone	string	O nome do fuso horário IANA (por exemplo `Europe/Madrid`) gravado com os eventos recorrentes e suas instâncias, o que o Google exige. Vazio (padrão) usa o fuso horário do calendário, lido uma vez.
Busy	Boolean	Public, somente leitura. True enquanto uma sincronização é executada.
LastSync-Time	TDateTime	Public, somente leitura. A hora local ao final da última sincronização bem-sucedida.
State	TsgcPlannerGoogleSyncState	Public, somente leitura. O estado da sincronização: token de sincronização, id do calendário, fuso horário e uma entrada por item mapeado. Altere-o somente quando Busy não estiver ativo.

Métodos

Método	Retorna	Descrição
Sync	Boolean	Executa uma sincronização no thread chamador (principal) e retorna quando ela termina. False quando não está Active, quando está Busy (OnError informa) ou quando a execução falhou (OnError informa o motivo).
SyncAsync	Boolean	Inicia uma sincronização em um thread de trabalho e retorna imediatamente. Data é alterado e cada evento é disparado no thread principal, por meio de Synchronize. False quando não está Active ou está Busy.
ResetState		Esquece o token de sincronização e os itens mapeados, de modo que a próxima sincronização seja completa. Ignorado enquanto Busy estiver ativo.
SaveStateToStream		Grava o estado como JSON (UTF-8), para aplicações que o guardam com seus próprios dados em vez de usar StateFileName.
LoadStateFromStream	Boolean	Lê um estado gravado por SaveStateToStream ou salvo em StateFileName. False, com o estado inalterado, quando o texto não é um estado deste formato e desta versão.

Eventos

Evento	Tipo	Descrição
OnConflict	TsgcPlannerGoogleConflictEvent	Um item alterado aqui e no Google desde a última sincronização: aItem é o item local e aEventJSON o evento como o Google o retornou. aResolution começa como gcrLastModified (a alteração mais recente vence, um empate fica com o Google). Defina gcrKeepLocal, gcrKeepRemote ou gcrSkip para deixar os dois lados como estão nesta execução (o item continua alterado). Uma edição contra uma exclusão não é informada aqui, a edição vence.

OnError	TsgcPlannerWSErrorEvent	aError informa por que uma sincronização falhou: um status HTTP, um erro de autorização ou uma sincronização que já está em execução. Também informa, uma vez por regra, uma regra de recorrência que o planner não consegue expandir; essa regra continua sendo sincronizada sem alteração.
OnFilterItem	TsgcPlannerGoogleFilterItemEvent	Chamado para cada item local antes de ser enviado. Defina aSync como False para manter o item fora do Google.
OnSyncDone	TsgcPlannerGoogleSyncDoneEvent	Uma sincronização terminou sem erro: aPulled eventos aplicados aqui, aPushed itens gravados no Google, aDeleted eventos excluídos no Google e aConflicts conflitos encontrados.

Threads

- SyncAsync e a sincronização automática falam com o Google em um thread de trabalho e alteram Data no thread principal, dentro de um único BeginUpdate e EndUpdate, de modo que as visões sejam redesenhadas uma só vez. Cada evento é disparado no thread principal.
- Enquanto Busy estiver ativo, apenas a sincronização em execução usa Client. Não chame o cliente a partir da aplicação nesse momento.
- Destruir o componente interrompe a sincronização automática e espera a que estiver em execução. Não libere o componente dentro dos seus próprios manipuladores de eventos.

Métodos protected

O transporte é um conjunto de métodos virtuais protected. Cada um retorna a resposta bruta e seu status HTTP e é executado no thread de trabalho (ou no thread principal para Sync). A implementação padrão chama o cliente em CalendarId. Um descendente pode sobrescrevê-los para testar a sincronização sem rede, como as demos fazem com a opção `--test`.

Método	Descrição
DoListEvents(const aPageToken, aSyncToken: string; out aStatus: Integer): string	Uma página da lista de eventos.
DoInsertEvent(const aJSON: string; out aStatus: Integer): string	Insere um evento.
DoUpdateEvent(const aEventId, aJSON: string; out aStatus: Integer): string	Aplica um patch a um evento. aJSON contém apenas os campos de que o planner é dono, de modo que os demais campos do evento são mantidos.
DoDeleteEvent(const aEventId: string; out aStatus: Integer): string	Exclui um evento. 404 e 410 contam como concluído.
DoGetCalendar(out aStatus: Integer): string	Lê o calendário, para o seu fuso horário, quando nenhuma lista de eventos o forneceu.
DoError(const aError: string)	Dispara OnError.

Exemplo

Um objeto de sincronização para um formulário ou um serviço: OAuth2 com um cliente de Aplicativo para computador, o estado salvo ao lado do executável, uma sincronização automática a cada cinco minutos e um log.

```
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;
```

Veja também

- Sincronização com o Google Agenda
- TsgcPlannerICalSubscription
- Sincronização com o Google Agenda e iCalendar no .NET
- TsgcPlannerData
- Recorrência
- Edições
- Demos

TsgcPlannerICalSubscription

Unit: sgcPlannerWS_ICal
TsgcPlannerICalSubscription assina, para um TsgcPlannerData, um feed iCalendar (.ics) somente leitura publicado em uma URL, como o endereço público ou secreto de um calendário do Google, Outlook ou Apple, um calendário de equipe ou uma lista de feriados. Ele baixa o feed com o cliente HTTP do sgcWebSockets, lê com TsgcPlannerICalendar e mescla seus eventos nos dados. Faz parte do pacote de ponte sgcPlannerWS (edição Registered, SGC_PLANNER_ADVANCED).

Download

- Uma URL `webcal://` ou `webcals://` é obtida como `https://`.
- O componente segue por conta própria até cinco redirecionamentos (301, 302, 303, 307 e 308), resolvendo os valores relativos de Location em relação à URL atual.
- Envia If-None-Match e If-Modified-Since com o ETag e o Last-Modified da última resposta importada. 304 Not Modified pula a importação. Uma mudança de URL ou de IdPrefix esquece esses validadores.
- Um corpo maior que MaxSize, qualquer outro status, um erro de transporte ou um feed sem BEGIN:VCALENDAR dispara OnError e deixa os itens e os validadores como estavam, de modo que a próxima atualização tenta de novo.

Mesclagem

- Cada item do feed recebe IdPrefix mais seu id: o UID do evento, ou UID#RECURRENCE-ID para uma ocorrência modificada. ParentId e RecurrenceParentId recebem o mesmo prefixo.
- Ids novos são adicionados, ids existentes cujos campos diferem são substituídos por inteiro, e os itens da assinatura que não estão mais no feed são excluídos, tudo em um único BeginUpdate e EndUpdate de Data.
- Um item pertence à assinatura quando seu Id começa com IdPrefix. Com IdPrefix vazio, os itens da última atualização pertencem a ela (mantidos apenas na memória). Os outros itens de Data nunca são tocados, de modo que itens locais e várias assinaturas, cada uma com seu próprio IdPrefix, compartilham um único objeto de dados.
- Os nomes RESOURCES do feed encontram os recursos de Data, e os novos recursos são adicionados a Data.
- As edições locais dos itens da assinatura são sobrescritas pela próxima atualização que importar. Vete-as em OnItemChanging de Data com IsSubscriptionItem, como mostra o exemplo.

Propriedades

Propriedade	Tipo	Descrição
Active	Boolean	Padrão False. True inicia um thread de trabalho que atualiza imediatamente e depois a cada RefreshInterval minutos (apenas uma vez quando RefreshInterval é 0).
Data	TsgcPlannerData	Os dados do planner nos quais o feed é mesclado.
IdPrefix	string	O prefixo dos ids dos itens do feed. Padrão `ics:` , que também é o ExcludeIdPrefix padrão de TsgcPlannerGoogleSync, de modo que os eventos do feed nunca são enviados ao Google.
MaxSize	Integer	O maior corpo aceito, em bytes. Padrão 10485760 (10 MB); 0 ou menos significa sem limite.

RefreshInterval	Integer	Minutos entre duas atualizações enquanto Active estiver ativo, 0 (padrão) significa uma única atualização.
TLSOptions	TsgcHTTPTLS_Options	As opções TLS do cliente HTTP interno, para feeds HTTPS.
URL	string	O endereço do feed: https://, http://, webcal:// ou webcals://.
Busy	Boolean	Public, somente leitura. True enquanto uma atualização é executada.
HTTPClient	TsgcHTTP1Client	Public, somente leitura. O cliente HTTP interno do sgcWebSockets, para proxy ou outras configurações. Enquanto Busy estiver ativo, apenas a atualização em execução o usa.
LastRefreshTime	TDateTime	Public, somente leitura. A hora local da última atualização bem-sucedida, 0 quando não houve nenhuma.

Métodos

Método	Retorna	Descrição
Refresh	Boolean	Baixa e importa o feed no thread chamador (principal). False quando Busy ou em caso de erro.
RefreshAsync	Boolean	Inicia uma atualização em um thread de trabalho e retorna imediatamente. A importação e cada evento são executados no thread principal. False quando Busy.
IsSubscriptionItem(aItem)	Boolean	True quando aItem pertence a esta assinatura.

Eventos

Evento	Tipo	Descrição
OnBeforeRefresh	TNotifyEvent	Uma atualização começa, antes do download.
OnError	TsgcPlannerWSErrorEvent	Uma atualização falhou: aError informa o status HTTP, o limite de tamanho, redirecionamentos demais, um erro de transporte ou um feed que não é analisado. Os itens permanecem como estavam.
OnImportWarning	TsgcPlannerICalWarningEvent	Uma parte do feed que TsgcPlannerICalendar ignorou ou leu de forma aproximada, com aLine e aMessage.
OnRefreshed	TsgcPlannerICalRefreshedEvent	Uma atualização teve êxito: aAdded, aUpdated e aDeleted itens. Uma resposta 304 Not Modified resulta em 0, 0 e 0.

Threads

Refresh bloqueia o thread chamador. RefreshAsync e o laço de Active baixam em um thread de trabalho e importam no thread principal por meio de Synchronize, de modo que cada evento é disparado no thread principal. Destruir o componente interrompe o thread de trabalho. Não libere o componente dentro dos seus próprios manipuladores de eventos.

Métodos protected

Método	Descrição
DoFetch(const aURL: string; aRequestHeaders, aResponseHeaders: TStrings; aBody: TStream): Integer	Um GET de aURL que não segue redirecionamentos. aRequestHeaders contém as linhas de cabeçalho da requisição (Nome: valor), aResponseHeaders recebe as linhas de cabeçalho da resposta e aBody o corpo. Retorna o status HTTP e levanta uma exceção em caso de erro de transporte. Chamado no thread da atualização; testes e demos o sobrescrevem para ler um arquivo local.
DoGetRefreshIntervalMs: Cardinal	O intervalo do laço de Active em milissegundos.
DoError, DoRefreshed	Disparam OnError e OnRefreshed.

Exemplo

Um calendário de equipe exibido ao lado dos itens locais, atualizado a cada hora, com os eventos do feed somente leitura nas visões.

```
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;
```

Veja também

- TsgcPlannerICalendar
- TsgcPlannerGoogleSync
- Sincronização com o Google Agenda
- iCalendar
- Sincronização com o Google Agenda e iCalendar no .NET
- TsgcPlannerData
- Recorrência
- Edições
- Demos

Visão geral do .NET

O **sgcPlanner .NET** é um port C# gerenciado da biblioteca Delphi sgcPlanner, não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcPlanner** e mantêm os nomes do Delphi, de modo que **TsgcPlannerData**, **TsgcPlannerItem**, **TsgcPlannerCalendar**, **TsgcPlannerRenderer** e as enumerações se leem igual nas duas linguagens. As três visões seguem o padrão de nomes do Delphi em todos os hosts: **TsgcVCLGantt** é **TsgcWinFormsGantt**, **TsgcWpfGantt** e **TsgcAvaloniaGantt**, e igualmente o Scheduler e o quadro Kanban. As visões são desenhadas com o port .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe.
Como no Delphi, um único **TsgcPlannerData** guarda os itens, os recursos, os status, as dependências e o calendário de trabalho, e qualquer número de visões o mostra. Uma edição em uma visão passa pelos dados, que notificam cada visão vinculada uma só vez, veja Dados e agendamento no .NET.

Pacotes

O port é dividido em cinco pacotes NuGet, de modo que um projeto traz apenas o host que usa.

PackageId	Contém	Frameworks de destino
esegece.sgcPlanner	Os dados, o calendário de trabalho, a escala de tempo, a recorrência, o motor de agendamento, os modelos de visão e os painters do Scheduler, da Timeline de recursos, do Gantt e do quadro Kanban, a fonte **System.Data.DataTable**, JSON, iCalendar e MS Project XML, e a saída SVG das visões. Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPlanner.Rendering	A saída PNG sem janela: **TsgcPlannerRenderer** e os métodos de extensão **SaveToPNG**, **ExportPNG** e **RenderToRaster** de **sgcPlanner_Raster**, desenhados no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, Linux e macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcPlanner.WinForms	**TsgcWinFormsScheduler**, **TsgcWinFormsGantt** e **TsgcWinFormsKanban** sobre **TsgcPlannerWinFormsView**.	net472, net8.0-windows, net9.0-windows
esegece.sgcPlanner.Wpf	**TsgcWpfScheduler**, **TsgcWpfGantt** e **TsgcWpfKanban** sobre **TsgcPlannerWpfView**.	net472, net8.0-windows, net9.0-windows
esegece.sgcPlanner.Avalonia	**TsgcAvaloniaScheduler**, **TsgcAvaloniaGantt** e **TsgcAvaloniaKanban** sobre **TsgcPlannerAvaloniaView**, no Windows, Linux e macOS com o Avalonia 11.	net8.0, net9.0

Os três pacotes de host dependem de esegece.sgcPlanner.Rendering, que depende de esegece.sgcPlanner, de modo que uma linha `PackageReference` traz tudo o que um host precisa. Veja Instalação pelo NuGet. A sincronização com o Google Agenda e a assinatura iCalendar da ponte sgcWebSockets estão no pacote separado **esegece.sgcPlanner.sgcWebSockets**, somente na edição Registered.

Edições

Edição	Ids dos pacotes	Inclui
Registered	esegece.sgcPlanner, .Rendering, .WinForms, .Wpf e .Avalonia	Tudo: as três visões nos três hosts, arrastar e soltar, resumos, marcos, progresso, temas, a fonte DataTable, JSON, a saída PNG e SVG sem janela e os recursos avançados: recorrência, o motor de agendamento com setas de dependência, o caminho crítico e as linhas de base, a Timeline de recursos, as swimlanes e os limites de WIP no quadro, iCalendar e MS Project XML.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcPlanner.WinForms.Community.	Tudo, exceto os recursos avançados. As propriedades permanecem, de modo que o mesmo código de formulário compila nas duas edições: um item recorrente aparece uma só vez no seu início, **ShowDependencies**, **AutoSchedule**, **ShowCriticalPath** e **ShowBaseline** ficam inertes, **Schedule** e **SetBaseline** do Gantt disparam **EsgcPlannerError**, **psvTimeline** pinta um aviso e **SwimlaneBy** age como **pslNone**. As classes das units avançadas não estão nos pacotes Community: **TsgcPlannerScheduleEngine**, **TsgcPlannerICalendar**, **TsgcPlannerMSProject**, **TsgcPlannerSchedulerTimelineView** e as classes de recorrência.

É a mesma divisão das edições do Delphi, veja Edições.

Hosts

Host	Contrapartida no Delphi	Plataformas
WinForms	VCL (controles TsgcVCL)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	VCL e FMX	Windows, .NET Framework 4.7.2 e .NET 8 e 9. Os controles podem ser declarados em XAML.
Avalonia	FMX (controles TsgcFMX)	Windows, Linux e macOS, .NET 8 e 9.
Sem janela	TsgcPlannerRenderer e os modelos de visão	Qualquer plataforma dos frameworks de destino: SVG com esegece.sgcPlanner, PNG com esegece.sgcPlanner.Rendering.

Todo host tem a mesma API: as propriedades, os métodos e os eventos dos controles Delphi com os mesmos nomes, mais **Data**, **Theme**, **Colors**, **ReadOnly**, **DragOptions**, **SyncSelection**, **ItemAt**, **ScrollToItem**, **SaveToPNG** e **SaveToSVG** na classe base. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja Também

- Instalação pelo NuGet
- Início Rápido WinForms
- Início Rápido WPF

- Início Rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Dados e agendamento no .NET
- Vínculo de dados no .NET
- JSON, iCalendar e MS Project no .NET
- Renderização sem janela no .NET
- Demos do .NET
- Plataformas

Instalação pelo NuGet

Os pacotes estão em nuget.org. Adicione o pacote do host; os pacotes core e de renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcPlanner.WinForms
dotnet add package esegece.sgcPlanner.Wpf
dotnet add package esegece.sgcPlanner.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcPlanner.WinForms.Community
dotnet add package esegece.sgcPlanner.Wpf.Community
dotnet add package esegece.sgcPlanner.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcPlanner.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie os pacotes Registered ou os Community em um projeto, nunca os dois: eles contêm os mesmos tipos no mesmo namespace.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:sgc="clr-namespace:esegece.sgcPlanner;assembly=esegece.sgcPlanner.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicação desktop Avalonia. As barras de rolagem das visões tiram sua aparência do tema Fluent. Mapeie o namespace no XAML com `xmlns:sgc="using:esegece.sgcPlanner"`.
- Sem janela: esegece.sgcPlanner sozinho basta para os dados, os formatos de arquivo e o SVG. Adicione esegece.sgcPlanner.Rendering para PNG; ele traz o SkiaSharp e seus recursos nativos para Windows e Linux.
- Sincronização com o Google Agenda e assinatura iCalendar: adicione **esegece.sgcPlanner.sgcWebSockets** (edição Registered), que depende de esegece.sgcPlanner e esegece.sgcWebSockets.

Em C# todo host precisa de `using esegece.sgcPlanner;`, e de `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUINotifyEvent** e os demais tipos do sgcUICore.

Veja Também

- Visão geral do .NET
- Início Rápido WinForms
- Início Rápido WPF
- Início Rápido Avalonia
- Demos do .NET

Início Rápido WinForms

Este início rápido monta um formulário em código com as três visões WinForms vinculadas a um único **TsgcPlannerData**: um gráfico de Gantt em cima, um quadro Kanban à esquerda e uma semana útil à direita. Arraste um cartão para outra coluna e a coluna de status do Gantt o acompanha.

Passos

1. Crie um projeto Windows Forms para `net8.0-windows` e adicione o pacote **esegece.sgcPlanner.WinForms** (**esegece.sgcPlanner.WinForms.Community** na edição Community).
2. Adicione `using esegece.sgcPlanner;` e `using esegece.sgcUICore;`.
3. Crie um **TsgcPlannerData** e preencha-o: status, recursos e itens.
4. Crie os controles, atribua **Data** e adicione-os ao formulário. São controles WinForms comuns: Dock, Anchor, TabIndex e Enabled funcionam como de costume.
5. Atribua os eventos. São os eventos dos controles Delphi, com os mesmos nomes.

Exemplo

Observações

- O quadro Kanban recebe uma coluna por status de **Data.Statuses** (**ColumnSource** é **pkcStatuses**). Defina **ColumnSource** como **pkcList** e preencha **Columns** para escolher as colunas e seus limites de WIP.
- A propriedade **View** do Scheduler é o tipo de visão (**psvDay**, **psvWorkWeek**, **psvWeek**, **psvMonth**, **psvAgenda**, **psvTimeline**), como no Delphi. O modelo de visão é **SchedulerView**; o Gantt e o quadro expõem os seus como **GanttView** e **KanbanView**.
- **Theme** é por controle. As barras de rolagem são as barras nativas do Windows, e **BorderStyle** escolhe a borda.
- **SaveToPNG** e **SaveToSVG** escrevem o que o controle mostra, no seu tamanho ou em outro tamanho e escala.

Veja Também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Dados e agendamento no .NET
- Início Rápido VCL

Início Rápido WPF

Os controles WPF podem ser declarados em XAML como qualquer outro controle. Este início rápido declara um Gantt, um quadro Kanban e um Scheduler em um grid e os vincula a um único **TsgcPlannerData** no code-behind.

Passos

1. Crie um projeto WPF para `net8.0-windows` e adicione o pacote **esegece.sgcPlanner.Wpf** (**esegece.sgcPlanner.Wpf.Community** na edição Community).
2. Mapeie o namespace na janela: `xmlns:sgc="clr-namespace:esegece.sgcPlanner;assembly=esegece.sgcPlanner.Wpf"`.
3. Declare os controles. As propriedades simples podem ser definidas como atributos, e as enumerações recebem os nomes dos seus valores, como `View="psvWorkWeek"`. Um controle sem Width e Height em um painel que não o estica pede 400 x 300.
4. Crie os dados no code-behind e atribua-os a **Data** de cada controle.

Exemplo

A janela:

```
<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>
```

O code-behind:

Observações

- As propriedades são propriedades CLR simples, não propriedades de dependência, portanto aceitam valores literais no XAML, mas não `{Binding}`. Os dados de um plano são vinculados por meio do **TsgcPlannerData**, veja Vínculo de dados no .NET.
- As barras de rolagem são filhos `ScrollBar` do WPF do controle. **IsTabStop** toma o lugar do **TabStop** do WinForms, e não há **BorderStyle**: use BorderBrush e BorderThickness ou um Border.
- Os controles executam um timer para a linha do horário atual enquanto estão carregados. **Dispose** para os timers e desanexa o listener dos dados; chame-o quando a janela for fechada.

Veja Também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET

- Vínculo de dados no .NET
- TsgcVCLGantt

Início Rápido Avalonia

Os controles Avalonia rodam no Windows, Linux e macOS. Este início rápido é uma aplicação Avalonia completa em código: um quadro Kanban ao lado de uma semana agrupada por recurso, ambos no tema escuro e vinculados a um único **TsgcPlannerData**.

Passos

1. Crie um projeto de console `net8.0`, defina `<OutputType>WinExe</OutputType>` e adicione **Avalonia.Desktop** e **Avalonia.Themes.Fluent** 11.3 e o pacote **esegece.sgcPlanner.Avalonia** (**esegece.sgcPlanner.Avalonia.Community** na edição Community).
2. Adicione o tema Fluent à aplicação.
3. Crie os dados e os controles e adicione os controles a um painel. No XAML, mapeie o namespace com `xmlns:sgc="using:esegece.sgcPlanner"`.

Exemplo

Observações

- **Theme** dos controles é o tema do sgcUICore. Ele oculta o **StyledElement.Theme** do Avalonia, portanto defina-o por meio de uma variável do tipo do controle, como no exemplo.
- **GroupByResource** dá a cada pessoa um grupo de colunas nas visões de dia, semana útil e semana, em todas as edições. A Timeline de recursos (**psvTimeline**) é um recurso Registered.
- As propriedades são propriedades CLR simples, não propriedades estilizadas: defina-as em código ou como atributos XAML.
- No macOS a tecla Command conta como Ctrl, portanto Command e a roda do mouse dão zoom no Gantt.

Veja Também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Dados e agendamento no .NET
- Início Rápido FireMonkey

Classes do .NET

Cada classe Delphi mantém o seu nome no .NET. Os controles VCL e FireMonkey se tornam os controles WinForms, WPF e Avalonia com o prefixo do host, e as classes independentes de framework são as mesmas em todos os hosts.

Classes dos hosts

Delphi	WinForms	WPF	Avalonia	Modelo (propriedade do host)
TsgcVCLScheduler, TsgcFMXScheduler	TsgcWinFormsScheduler	TsgcWpfScheduler	TsgcAvaloniaScheduler	TsgcPlannerSchedulerView (SchedulerView): TsgcPlannerSchedulerTimelineView na edição Registered, TsgcPlannerSchedulerRenderView na edição Community
TsgcVCLGantt, TsgcFMXGantt	TsgcWinFormsGantt	TsgcWpfGantt	TsgcAvaloniaGantt	TsgcPlannerGanttRenderView (GanttView)
TsgcVCLKanban, TsgcFMXKanban	TsgcWinFormsKanban	TsgcWpfKanban	TsgcAvaloniaKanban	TsgcPlannerKanbanRenderView (KanbanView)
TsgcPlannerVCLView, TsgcPlannerFMXView	TsgcPlannerWinFormsView	TsgcPlannerWpfView	TsgcPlannerAvaloniaView	TsgcPlannerView (View)

As classes base derivam de **TsgcUIWinFormsControl**, **TsgcUIWpfControl** e **TsgcUIAvaloniaControl** do sgcUICore.

Membros de cada controle

Membro	Descrição
Data	O **TsgcPlannerData** que o controle mostra. Vários controles podem compartilhar um.
View	O modelo de visão, um **TsgcPlannerView**. As propriedades tipadas SchedulerView, GanttView e KanbanView retornam o mesmo objeto. No Scheduler, **View** é redeclarada como o tipo de visão, como no Delphi.
Theme, Colors	A aparência: o **TsgcUITheme** (utLight, utDark) e as **TsgcPlannerViewColors** que substituem as cores do tema.
ReadOnly, DragOptions	Desativa a edição ou escolhe os arrastes que o usuário pode fazer (criar, mover, redimensionar, progresso, vincular, reordenar), como `[Flags]` **TsgcPlannerDragOptions**.
SyncSelection	A seleção segue o **SelectedId** dos dados, de modo que as visões vinculadas aos mesmos dados selecionam o mesmo item.
ItemAt, ScrollToItem	O item em um ponto do controle (pixels do cliente no WinForms, DIPs no WPF e no Avalonia) e a rolagem de um item até ficar visível.
UpdateScrollBars, CursorKind	Atualizam as barras de rolagem depois de uma alteração feita no modelo, e o cursor que o controle mostra.

OnItemClick, OnItemDblClick, OnItemDelete, OnSelection-Change	Os eventos da visão base do Delphi. OnItemDelete pode recusar a exclusão.
SaveToPNG, SaveToSVG	Desenham a visão como ela está agora em um stream ou em um arquivo, no tamanho do controle ou em outro tamanho, com uma escala.

Tipos independentes de framework

Tipo	Descrição
TsgcPlannerData	Itens, recursos, status, dependências, o calendário de trabalho, a notificação de alterações, JSON. Veja TsgcPlannerData.
TsgcPlannerItem, TsgcPlannerResource, TsgcPlannerStatus, TsgcPlannerDependency	Os itens das coleções, em **TsgcPlannerItems**, **TsgcPlannerResources**, **TsgcPlannerStatuses** e **TsgcPlannerDependencies**.
TsgcPlannerCalendar, TsgcPlannerHolidays	Os dias úteis, os horários de trabalho e os feriados. Veja Calendário de Trabalho.
TsgcPlannerTimeScale	O zoom e o ajuste do Gantt e da Timeline. Veja Escala de Tempo.
TsgcPlannerSchedulerView, TsgcPlannerGanttView, TsgcPlannerKanbanView	Os modelos de visão com suas subclasses de renderização: layout, hit testing, teclado, mouse e arraste, e **ExportImage** para SVG. Não precisam de janela.
TsgcPlannerScheduleEngine	O motor de agendamento (Registered). Veja Agendamento Automático.
TsgcPlannerSource_DataSet	Vincula os dados a tabelas **System.Data.DataTable**. Veja Vínculo de dados no .NET.
TsgcPlannerICalendar, TsgcPlannerMSProject	Importação e exportação de iCalendar e MS Project XML (Registered). Veja JSON, iCalendar e MS Project no .NET.
TsgcPlannerRenderer	Em esegece.sgcPlanner.Rendering: as três visões em PNG e SVG sem janela. Veja Renderização sem janela no .NET.
TsgcPlannerLocale	Os nomes dos dias e dos meses, o formato de hora e o separador decimal dos rótulos do cabeçalho, capturados de `CultureInfo.CurrentCulture`.
sgcPlanner_DateUtils, sgcPlanner_JSON, sgcPlanner_Raster	Classes estáticas com as funções livres do Delphi: as conversões de tempo e o relógio, as funções JSON e os métodos de extensão PNG das visões em esegece.sgcPlanner.Rendering.

Enumerações

As enumerações mantêm os nomes e os prefixos de valor do Delphi, por exemplo **TsgcPlannerSchedulerViewKind** (psv), **TsgcPlannerGanttField** (pgc), **TsgcPlannerBarColorMode** (pbc), **TsgcPlannerDurationUnit** (pdu), **TsgcPlannerDependencyKind** (pdk), **TsgcPlannerConstraint** (pc), **TsgcPlannerKanbanColumnSource** (pkc), **TsgcPlannerSwimlaneBy** (psl), **TsgcPlannerWIPMode** (pwm), **TsgcPlannerRecurrenceScope** (prs) e **TsgcPlannerRendererViewKind** (prv). Os sets do Delphi são enumerações `[Flags]` combinadas com `|`: **TsgcPlannerWeekDays**, **TsgcPlannerChangeKinds**, **TsgcPlannerItemFields** e **TsgcPlannerDragOptions**. O **TAlignment** da RTL do Delphi, usado nas colunas do Gantt, é declarado em esegece.sgcPlanner.

Veja Também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- Arquitetura

Diferenças do .NET em relação ao Delphi

O port .NET mantém os nomes, os padrões, o comportamento do teclado e do mouse e as regras de layout do Delphi. A saída SVG das visões é idêntica byte a byte à do Delphi, e a saída PNG coincide com ela dentro da tolerância do rasterizador SkiaSharp. As diferenças abaixo vêm dos frameworks dos hosts e da biblioteca de classes do .NET.

Tempos e valores

- As propriedades de data são valores `DateTime` de Kind Unspecified: **Start**, **Finish**, **ConstraintDate**, **BaselineStart**, **BaselineFinish**, **RecurrenceDate** e **ProjectStart**. O TsgcPlannerTime do Delphi, em minutos inteiros, é um `long`, acessado pelas propriedades com o sufixo Time: **StartTime**, **FinishTime** e assim por diante. **Duration** é em minutos.
- O relógio é `sgcPlanner_DateUtils.sgcPlannerNow`, um `Func<DateTime>`. Substitua-o para congelar o tempo nos testes e restaure depois o delegate salvo.
- TsgcUIColor é um `uint`, $AARRGGBB, em que 0 significa a cor do tema ou do status.
- TFormatSettings e as globais da RTL são substituídos por `CultureInfo.CurrentCulture`, capturada pelo **TsgcPlannerLocale**. As propriedades indexadas de nomes de dias e meses são os métodos **GetShortDayName**, **SetShortDayName**, **GetLongMonthName** e assim por diante. Os arquivos são sempre escritos com formatos invariantes.

Eventos, métodos e listas

- Os eventos são propriedades delegate com os mesmos nomes do Delphi (**OnChange**, **OnItemCreate**, **OnCardMoving**...). Atribua um só handler com `=`. Um parâmetro `var` do Delphi é um parâmetro `ref`, e uma lambda com um parâmetro `ref` declara o tipo de todos os parâmetros. **TNotifyEvent** é **TsgcUINotifyEvent**.
- Os dados aceitam vários listeners: **AddListener** e **RemoveListener** recebem um **TsgcPlannerChangeEvent**, como as visões.
- Uma propriedade indexada do Delphi é um método: `Rows(i)`, `ColumnInfo(i)`, `Cards(i)`, `Entries(i)`, `Calendar.Intervals(i)`. Uma função do Delphi sem parâmetros continua sendo um método, como `RowCount()` ou `WorkingMinutesPerDay()`.
- Um array `var` do Delphi é um array `ref` que o método substitui por outro do tamanho exato: `ChildrenOf(id, ref aList)`, `GetOccurrences(start, finish, ref aList)`.
- Um set do Delphi é uma enumeração `[Flags]` combinada com `|`. As coleções mantêm a forma de TCollection: Add, Delete, Clear, Assign, BeginUpdate, EndUpdate, um indexador e `foreach`.

Sem streaming de componentes e sem notificação de liberação

- Não há DFM no .NET. Os controles são criados em código ou em XAML, e os dados, as colunas e as opções são definidos em código.
- O .NET não tem notificação de liberação. Uma visão mantém seus dados até que **Data** seja definido como null ou o controle seja descartado. Os controles WPF e Avalonia e os modelos de visão são `IDisposable`: **Dispose** para seus timers e desanexa o listener dos dados. O Avalonia não descarta controles, portanto descarte-os quando a janela for fechada.
- Os dados não têm estado de carregamento: as alterações são acumuladas entre **BeginUpdate** e **EndUpdate** e notificadas uma só vez.

Hosts

- O WinForms usa as barras de rolagem nativas e o **BorderStyle** do WinForms; WPF e Avalonia têm filhos `ScrollBar` e não têm BorderStyle.
- WPF e Avalonia têm **IsTabStop** no lugar de **TabStop**. No Avalonia a tecla Command conta como Ctrl, e **Theme** oculta o **StyledElement.Theme** do Avalonia.

- As propriedades dos controles são propriedades CLR simples em todos os hosts, não propriedades de dependência nem estilizadas.
- O toque chega como entrada de mouse; o arraste por toque do Delphi não tem um caminho de código separado.
- O cursor de divisão (**pcrSplit**) é um cursor de redimensionamento horizontal no WPF e no Avalonia, que não têm cursor de divisão.

Fontes de dados e arquivos

- O **TsgcPlannerSource_DataSet** vincula tabelas `System.Data.DataTable` no lugar de TDataSet. Veja Vínculo de dados no .NET para as diferenças.
- O MS Project XML é lido com `System.Xml`. A unit sgcPlanner_XML do Delphi não foi portada e não é necessária.
- O **ExportImage** de um modelo de visão escreve apenas SVG. O PNG precisa de esegece.sgcPlanner.Rendering, veja Renderização sem janela no .NET.

Veja Também

- Visão geral do .NET
- Classes do .NET
- Dados e agendamento no .NET
- Vínculo de dados no .NET

Dados e agendamento no .NET

O **TsgcPlannerData** é a única fonte da verdade de um plano. Ele guarda os itens (tarefas, eventos e marcos), os recursos, os status, as dependências e o calendário de trabalho. As visões nunca gravam um campo de item por conta própria: cada drop chama os dados dentro de um único **BeginUpdate** e **EndUpdate**, e os dados notificam cada listener uma só vez, de modo que um Gantt, um quadro e um Scheduler vinculados aos mesmos dados sempre concordam.

Montando um plano

O exemplo monta um pequeno projeto com três status, duas pessoas, um calendário com dois turnos e um feriado, quatro tarefas ligadas de término a início e um marco, depois escuta as alterações, move um cartão como faz um drop do Kanban e executa o motor de agendamento.

Observações

- **Finish** é exclusivo, também para itens de dia inteiro: um item de dia inteiro em 1 de outubro vai de 1 de outubro 00:00 a 2 de outubro 00:00. Um item com **Start** igual a **Finish** é um marco.
- **Resources** e **Tags** são listas separadas por vírgulas; **ResourceAt**, **ResourceCount**, **TagAt** e **TagCount** as leem uma a uma. **ParentId** monta a árvore de tarefas e **UpdateSummaries** recalcula as tarefas de resumo.
- **MoveItem** altera o status e a posição de um cartão, **DeleteItem** exclui um item e os itens de exceção da sua série, e **SelectedId** é a seleção compartilhada pelas visões.
- **OnItemChanging** pode recusar uma edição antes de ela ser feita, para os campos de um conjunto **TsgcPlannerItemFields**.
- O motor posiciona cada tarefa que não é **Manual** o mais cedo possível depois dos seus vínculos e da sua **Constraint**, no calendário de trabalho, e calcula **TotalSlack** e **Critical**. Um ciclo de vínculos dispara **OnScheduleError** com os ids do ciclo. Um Gantt com **AutoSchedule** o executa após cada edição, **Schedule** do Gantt o executa sob demanda, e **SetBaseline** guarda as datas atuais como linha de base. O motor é um recurso Registered.

Veja Também

- Visão geral do .NET
- Classes do .NET
- JSON, iCalendar e MS Project no .NET
- Dados Compartilhados
- Agendamento Automático
- Calendário de Trabalho

Vínculo de dados no .NET

O **TsgcPlannerSource_DataSet** do Delphi carrega um plano de tabelas TDataSet e grava as edições de volta. No .NET a mesma classe vincula tabelas `System.Data.DataTable`, as tabelas que um `DataAdapter` de qualquer provedor ADO.NET preenche. As visões continuam vinculadas ao **TsgcPlannerData**, de modo que a mesma fonte atende um Gantt, um quadro e um Scheduler ao mesmo tempo.

Propriedade	Descrição
Data	O **TsgcPlannerData** que a fonte preenche.
DataSet, ResourceDataSet, DependencyDataSet	As tabelas dos itens, dos recursos e dos vínculos. Somente a tabela de itens é obrigatória.
ItemFields, ResourceFields, DependencyFields	Os nomes das colunas. Os padrões são os nomes das propriedades dos itens (Id, Title, Start, Finish, AllDay, Status, Progress, Color, ParentId, Tags, Resources, Rank, Notes, RecurrenceRule, DependsOn), dos recursos (Id, Name, Color, Initials) e dos vínculos (FromId, ToId, Kind, Lag). Uma coluna que não existe não é mapeada.
Active	Carrega as tabelas nos dados e as mantém sincronizadas enquanto for true.
LiveUpdates	As linhas adicionadas, alteradas ou excluídas em uma tabela chegam aos dados imediatamente (true por padrão).
WriteBack	As edições feitas nas visões ou no código são gravadas de volta nas linhas (true por padrão).
OnLoadItem	Disparado para cada item carregado, com a tabela e o `DataRow`, para preencher campos que o mapa não cobre.

Exemplo

Diferenças em relação ao TDataSet

- Um `DataTable` não tem cursor. Um Edit e Post do Delphi é um `DataRow.BeginEdit` e `EndEdit`, ou uma atribuição direta à coluna, e Cancel é `CancelEdit`. Uma linha em edição é gravada de volta quando a edição termina.
- A fonte escuta os eventos da tabela: uma linha adicionada ou alterada é um registro postado, uma linha excluída ou uma tabela limpa remove os itens, uma alteração rejeitada recarrega o registro, e `AcceptChanges` é ignorado. As linhas excluídas e desanexadas são ignoradas.
- As colunas graváveis são as que não são ReadOnly, AutoIncrement nem calculadas. Uma coluna de data numérica guarda datas OLE Automation, e um texto vazio atribuído a uma coluna que não é de texto define `DBNull`.
- O **OnLoadItem** passa o `DataRow`, porque não há registro atual. Os nomes das colunas são **GetName** e **SetName** por índice, além das propriedades nomeadas.
- Vincule somente objetos `DataTable`: um `DataView` ou um `IDataReader` não é uma fonte. Uma tabela descartada se desvincula sozinha, e a fonte é `IDisposable`.

Vinculando os controles

As propriedades dos controles são propriedades CLR simples, não propriedades de dependência nem estilizadas, porque um plano é um modelo inteiro e não um único valor. Em uma aplicação MVVM mantenha o **TsgcPlannerData** no view model e atribua-o a **Data** dos controles, e reaja a **OnChange** ou a um listener dos dados, a **SelectedId** e aos eventos dos controles.

Veja Também

- Visão geral do .NET
- Dados e agendamento no .NET
- Diferenças do .NET em relação ao Delphi
- TsgcPlannerSource_DataSet
- Vínculo com DataSet

JSON, iCalendar e MS Project no .NET

Um plano é salvo e carregado em três formatos, com os mesmos resultados do Delphi: os arquivos que uma aplicação .NET escreve são lidos por uma aplicação Delphi e vice-versa.

Formato	Classes	Edição
JSON, o formato nativo com todos os campos	**SaveToJSON**, **LoadFromJSON**, **SaveToFile** e **LoadFromFile** do **TsgcPlannerData**, e as funções de stream e de arquivo de **sgcPlanner_JSON**.	Ambas
iCalendar (RFC 5545), para Outlook, Google Agenda e Apple Calendar	**TsgcPlannerICalendar**	Registered
Microsoft Project XML (MSPDI): tarefas, vínculos, recursos, atribuições e o calendário	**TsgcPlannerMSProject**	Registered

Exemplo

Observações

- O JSON é escrito em UTF-8 sem marca de ordem de bytes; uma marca é aceita ao carregar. Um texto que não é UTF-8 válido dispara **EsgcPlannerError**.
- **TsgcPlannerICalendar** e **TsgcPlannerMSProject** são classes simples: crie-as, defina **Data** e chame **ImportFromString**, **ImportFromStream** ou **ImportFromFile**, e **ExportToString**, **ExportToStream** ou **ExportToFile**. **ImportMode** substitui os dados ou mescla neles os itens importados.
- Um problema na entrada não interrompe uma importação: ele dispara **OnImportWarning** com a linha e uma mensagem, e **WarningCount** os conta. As mensagens são constantes públicas das classes, com marcadores `{0}`.
- **ExportUTC** escreve os horários do iCalendar em UTC. **ProductId** é o PRODID da exportação.
- O MS Project XML é lido com `System.Xml.XmlReader`, com o DTD ignorado e sem resolvedor externo, e escrito pelo port do gravador do Delphi, de modo que a exportação é idêntica byte a byte à do Delphi.

Veja Também

- Visão geral do .NET
- Dados e agendamento no .NET
- JSON
- iCalendar
- XML do MS Project

Renderização sem janela no .NET

Cada visão desenha por meio do seu modelo de visão, e os modelos não precisam de janela. Um serviço, um teste ou um relatório pode carregar um plano e salvar o Scheduler, o Gantt e o quadro como PNG ou SVG em qualquer plataforma, inclusive Linux.

- O **TsgcPlannerRenderer** em esegece.sgcPlanner.Rendering possui as três visões. Defina **Data**, **View** (**prvScheduler**, **prvGantt**, **prvKanban**), **Width**, **Height**, **Scale**, **Theme** e as opções da visão, e chame **SaveToPNG** ou **SaveToSVG**.
- **SchedulerOptions**, **GanttOptions** e **KanbanOptions** têm as propriedades dos controles. **Assign** as copia de um modelo de visão, de modo que um relatório pode desenhar o que um controle mostra. O que as opções não cobrem é definido em **SchedulerView**, **GanttView** e **KanbanView**.
- Um modelo de visão também pode ser desenhado sozinho: **ExportImage** escreve SVG com esegece.sgcPlanner, e os métodos de extensão **SaveToPNG**, **ExportPNG** e **RenderToRaster** de **sgcPlanner_Raster** escrevem PNG.

Exemplo

Observações

- O tamanho é em pixels lógicos; uma escala 2 escreve uma imagem de 2400 x 1000 para uma visão de 1200 x 500. **DeviceWidth** e **DeviceHeight** retornam o tamanho da imagem.
- **FontFile** e **FontFileBold** escolhem um arquivo de fonte TrueType, para um servidor sem a fonte padrão.
- Os rótulos do cabeçalho seguem o **TsgcPlannerLocale** compartilhado de `sgcPlanner_DateUtils.sgcPlannerLocale()`, capturado da cultura atual no primeiro uso. Para obter a mesma saída em todas as máquinas, defina a cultura antes da primeira renderização, ou defina os nomes desse locale.
- Os controles também têm **SaveToPNG** e **SaveToSVG**, com o tamanho do controle.

Veja Também

- Visão geral do .NET
- Classes do .NET
- TsgcPlannerRenderer
- Renderização sem Janela

Sincronização com o Google Agenda e iCalendar no .NET

Pacote: esegece.sgcPlanner.sgcWebSockets (edição Registered), namespace esegece.sgcPlanner. Tem como destino netstandard2.0, net8.0 e net9.0, e usa a biblioteca .NET do sgcWebSockets (esegece.sgcWebSockets) para HTTP e OAuth2 do Google.
O pacote .NET tem os mesmos dois componentes da ponte Delphi, com os mesmos nomes, o mesmo mapeamento, a mesma regra de conflito e o mesmo formato de arquivo de estado, de modo que um arquivo de estado gravado pelo Delphi é carregado no .NET e vice-versa.

- **TsgcPlannerGoogleSync** mantém um TsgcPlannerData em sincronização bidirecional com um Google Agenda por meio de um TsgcHTTP_Google_Cloud_Calendar_API_Client (sua propriedade Client). Consulte TsgcPlannerGoogleSync.
- **TsgcPlannerICalSubscription** mescla nos dados uma URL iCalendar somente leitura (https://, http:// ou webcal://), analisada com TsgcPlannerICalendar. Consulte TsgcPlannerICalSubscription.

Diferenças em relação ao Delphi

- As duas classes são classes simples que implementam IDisposable, não componentes. Dispose interrompe o thread de trabalho; depois dele nada é aplicado.
- Os eventos são propriedades delegate, por exemplo OnRefreshed, OnError, OnImportWarning e OnBeforeRefresh (um EventHandler). OnConflict e OnFilterItem recebem seu resultado por ref.
- SyncAsync retorna Task<bool>. RefreshAsync retorna bool, False quando uma atualização está em execução.
- O trabalho que toca Data e dispara os eventos é executado no SynchronizationContext do thread que chamou SyncAsync ou RefreshAsync, ou definiu Active (o thread de interface no WinForms, WPF e Avalonia). Sem contexto, é executado no thread de trabalho.
- LastSyncTime e LastRefreshTime são DateTime; default(DateTime) significa nunca.
- O ponto de encaixe de transporte de TsgcPlannerICalSubscription é o método virtual protected DoFetch(string aURL, IList<string> aRequestHeaders, IList<string> aResponseHeaders, Stream aBody), que retorna o status HTTP. A implementação padrão usa o HTTPClient interno (TsgcHTTP1Client) com HandleRedirects desativado; o componente segue por conta própria até cinco redirecionamentos e envia If-None-Match e If-Modified-Since com os validadores da última importação.
- Não há notificações de liberação: destruir Data não limpa a propriedade Data.

Eventos de feed somente leitura

Uma atualização sobrescreve as edições locais dos itens da assinatura. Vete-as em OnItemChanging, que as visões consultam antes de confirmar um arraste, um redimensionamento ou uma edição:

```
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
```

Sincronização com o Google Agenda

```
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();
```

Os itens cujo Id começa com ExcludeIdPrefix (ics: por padrão, o IdPrefix de TsgcPlannerICalSubscription) nunca são enviados ao Google.

Demo

A demo WinForms demos\Planner\GoogleCalendarSync mostra os dois componentes em um TsgcWinFormsScheduler. Execute-a com --test para um autoteste offline: um transporte Google de teste e um servidor iCalendar em loopback, sem precisar de credenciais.

Veja também

- TsgcPlannerGoogleSync
- TsgcPlannerICalSubscription
- Sincronização com o Google Agenda
- TsgcPlannerICalendar
- TsgcPlannerData
- iCalendar

Demos do .NET

Os demos estão em `.net\demos\Planner`, uma pasta por demo com seu projeto, seu `Program.cs` e um README.md. Eles montam a interface em código, sem XAML. Cada demo tem uma opção `--test` que executa uma autoverificação fora da tela, escreve linhas PASS e FAIL e termina com o número de verificações que falharam, como a opção `/test` dos demos Delphi, e os demos com janela têm uma opção `--snapshot` que renderiza as visões em arquivos PNG.

Demo	Host	Mostra
ProjectPlanner	WinForms	Um projeto de relançamento de site mostrado de três formas ao mesmo tempo: um gráfico de Gantt, um quadro Kanban e uma semana, todos vinculados aos mesmos dados. Quatorze tarefas em duas trilhas, cinco pessoas, quatro status e vínculos de término a início. Arraste um cartão no quadro e o Gantt e o Scheduler o acompanham.
ProjectPlannerWpf	WPF	A mesma janela com os controles **TsgcWpf**.
ProjectPlannerAvalonia	Avalonia	A mesma janela com os controles **TsgcAvalonia**, para Windows, Linux e macOS.
TeamScheduler	WinForms	Três semanas de reuniões de equipe para cinco pessoas: dia, semana útil, semana, mês, agenda e a Timeline de recursos, uma reunião diária em pé e uma sincronização semanal como séries recorrentes, uma conferência de dia inteiro, arrastar para criar com **OnItemCreate** e agrupamento por recurso.
TeamSchedulerWpf	WPF	A mesma janela com **TsgcWpfScheduler**.
TeamSchedulerAvalonia	Avalonia	A mesma janela com **TsgcAvaloniaScheduler**.
HeadlessRender	Console	Sem janela: carrega um projeto de JSON (ou monta um e o salva como project.json) e escreve um PNG e um SVG do Scheduler, do Gantt e do quadro com **TsgcPlannerRenderer**, no Windows, Linux e macOS. A contrapartida .NET do demo HeadlessReport do Delphi.
GoogleCalendarSync	WinForms	Uma visão semanal mantida em sincronização bidirecional com uma agenda do Google, mais uma assinatura iCalendar somente leitura, com os componentes da ponte esegece.sgcPlanner.sgcWebSockets (Registered).

A parte sem interface de ProjectPlanner e TeamScheduler (os dados gerados, a configuração das visões, o snapshot e a autoverificação) está em `ProjectPlanner_Shared.cs` e `TeamScheduler_Shared.cs`, que os projetos WinForms, WPF e Avalonia compilam, de modo que os três hosts executam as mesmas verificações.
Cada demo compila e executa nas duas edições. Na edição Community o TeamScheduler não tem a entrada Timeline e mostra cada série uma só vez, o ProjectPlanner não desenha setas de dependência, e o GoogleCalendarSync explica que a ponte é um pacote Registered.

Compilando os demos

Dentro da árvore do produto os demos compilam contra os projetos da biblioteca. O zip dos demos usa por padrão os pacotes NuGet da edição Community. Compile um demo com os pacotes Registered definindo **SgcPlannerEdition**:

```
dotnet build ProjectPlanner\ProjectPlanner.csproj -p:SgcPlannerEdition=Registered
dotnet run --project HeadlessRender\HeadlessRender.csproj -- --test
```

Veja Também

- Visão geral do .NET
- Instalação pelo NuGet
- Início Rápido WinForms
- Demos

Solução de Problemas

EsgcPlannerError: This feature requires the Registered edition of sgcPlanner

Schedule e SetBaseline do Gantt o geram em um build Community. Coloque essas chamadas dentro de `{$IFDEF SGC_PLANNER_ADVANCED}`, ou use a edição Registered ou Trial. Veja Recursos por Edição.

Um formulário não carrega em um build Community: Class TsgcPlannerICalendar not found

TsgcPlannerICalendar e TsgcPlannerMSProject não fazem parte da edição Community. Crie-os no código dentro de `{$IFDEF SGC_PLANNER_ADVANCED}` em vez de soltá-los em um formulário que também precise carregar na Community.

O Scheduler não mostra eventos

- Verifique se os eventos caem no período em tela. Date 0 segue o relógio, de modo que a visão mostra a semana atual; defina Date ou chame GoToday, Next e Prior.
- As visões de dia mostram as horas de Options.DayStartHour a Options.DayEndHour. Um evento fora delas é desenhado cortado na borda.
- Um evento cujo Finish é anterior ao seu Start é lido como um evento de duração zero no seu Start.
- Certifique-se de que a propriedade Data do Scheduler é o objeto de dados que o seu código preenche. Um controle sem Data usa um objeto de dados interno próprio.

Um evento recorrente aparece apenas uma vez

A recorrência faz parte da edição Registered. Na Registered, verifique a regra com TsgcPlannerRecurrence.TryParse, que retorna o motivo quando uma regra não é suportada, e verifique COUNT, UNTIL e RecurrenceExceptions. Veja Recorrência.

O agendamento automático não move uma tarefa

- O mecanismo mantém as datas de tarefas Manual, itens recorrentes e tarefas de um ciclo de dependência. OnScheduleError lista as tarefas de um ciclo e Schedule retorna False.
- Uma tarefa com Constraint pcSNET não começa antes do seu ConstraintDate. Um movimento no Gantt define essa restrição; volte Constraint para pcASAP para deixar o mecanismo puxar a tarefa para mais cedo.
- AutoSchedule roda somente após uma edição feita no Gantt. Após uma alteração no código ou em outra visão, chame Schedule.

Veja Agendamento Automático.

O Gantt não desenha setas de dependência

As setas exigem a edição Registered e ShowDependencies True. Um vínculo só é desenhado quando as suas duas tarefas estão em linhas visíveis, de modo que um vínculo para um resumo recolhido fica oculto, e um vínculo com FromId ou ToId desconhecido nunca é desenhado.

Um arrasto não faz nada ou o item volta

Verifique ReadOnly e DragOptions do controle, e os handlers que podem recusar uma edição: OnItemChanging dos dados, OnCardMoving, OnItemCreate, OnLinkCreating e OnRecurrenceEdit. No quadro, uma coluna no seu limite de WIP com WIPMode pwmBlock recusa novos cartões, e a coluna desconhecida nunca é um destino de drop. No Scheduler, os itens de resumo são mostrados mas não podem ser arrastados. Veja Arrastar e Soltar.

EsgcPlannerError: OnChange kept changing the planner data for more than 8 rounds

Um handler de OnChange, ou um listener, altera os dados a cada notificação, de modo que as alterações nunca se estabilizam. Altere os dados somente quando diferem do valor desejado, ou tire a alteração do handler.

Os horários ficam algumas horas errados após uma importação ou exportação iCalendar

Horários em UTC ou com fuso horário são convertidos para o horário local da máquina na importação. Na exportação, ExportUTC grava horários UTC para os itens que não se repetem, que um calendário mostra no seu próprio fuso horário. Defina ExportUTC como False para gravar horários locais. Veja iCalendar.

Datas, durações ou nomes aparecem diferentes em outra máquina ou no Linux

Os rótulos e os números das durações usam sgcPlannerLocale, que é inicializado a partir da máquina. Defina as suas propriedades, por exemplo ShortDateFormat, Use24Hour e DecimalSeparator, na inicialização para ter a mesma saída em todo lugar.

Caracteres são perdidos no Delphi 7 ou no Delphi 2007

Essas versões têm strings Ansi, de modo que o texto UTF-8 de arquivos JSON, iCalendar e XML do MS Project é convertido para a página de código do sistema, e os caracteres fora dela são perdidos. O Delphi 2009 e posteriores mantêm todos os caracteres.

As edições não são gravadas de volta no dataset

- Verifique se WriteBack é True e se IdField nomeia um campo que Locate consegue encontrar.
- Nada é gravado enquanto o dataset está em estado de edição ou inserção; as alterações são gravadas após Post ou Cancel.
- Recursos, campos calculados, campos de lookup e campos somente leitura nunca são gravados.

Veja Vínculo com DataSet.

LoadFromJSON gera EsgcPlannerError

O texto não é JSON válido, não é um documento sgcPlanner, ou foi gravado com uma versão mais nova do formato. Os dados ficam como estavam, de modo que a aplicação pode relatar a mensagem e continuar a execução. Veja Persistência JSON.

Uma violação de acesso depois que um formulário ou um objeto é liberado

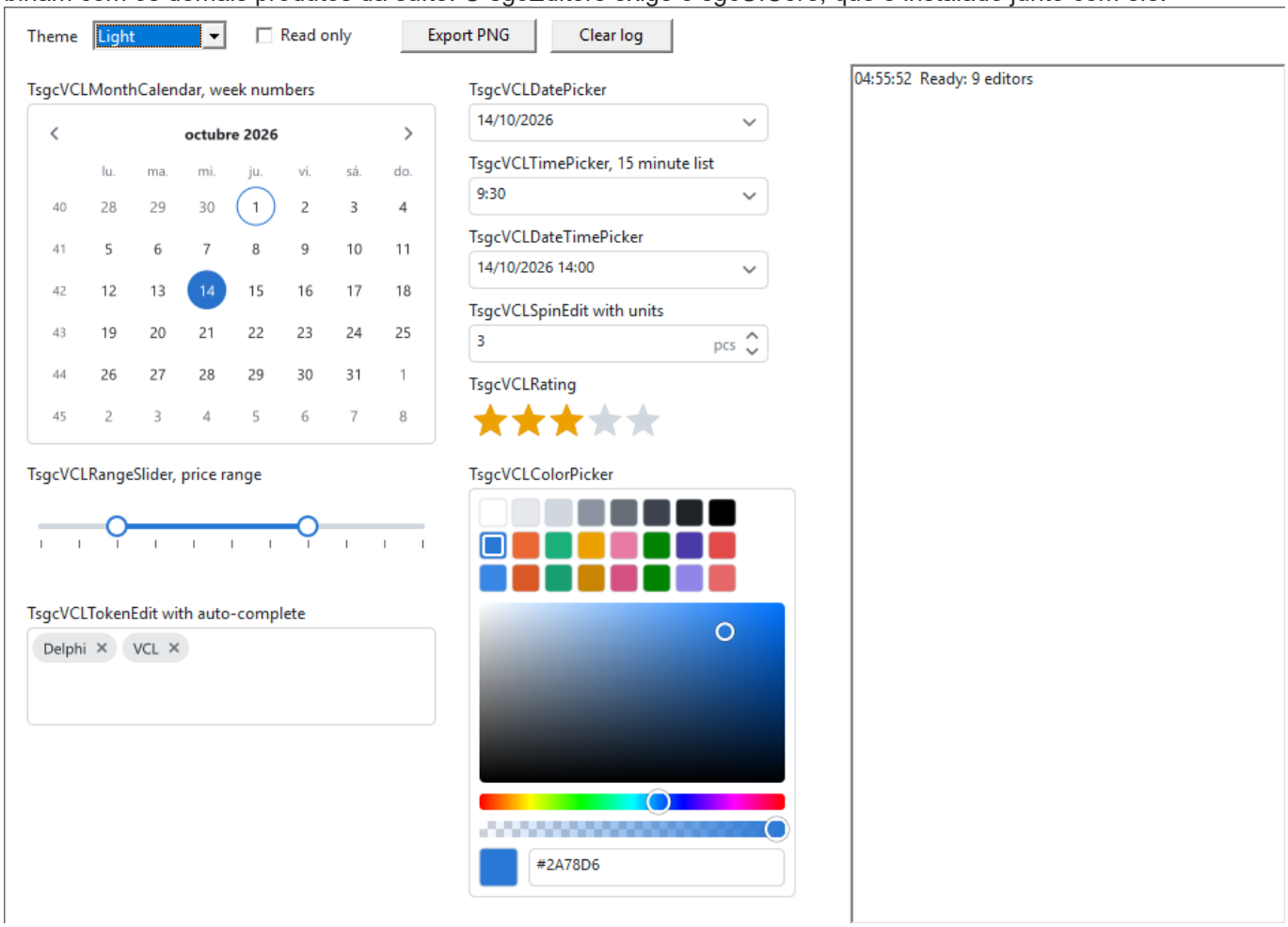
Um handler adicionado com AddListener deve ser removido com RemoveListener antes que o objeto que o possui seja liberado. Os dados não sabem quando esse objeto desaparece.

Veja Também

- Recursos por Edição
- Plataformas
- TsgcPlannerData

Introdução

sgcEditors adiciona os controles de entrada de um formulário a aplicativos Delphi e C++Builder, para VCL e FireMonkey: um calendário mensal, um seletor de data, um seletor de hora, um seletor de data e hora, um controle deslizante de intervalo, um seletor de cor, um token edit, uma avaliação e um spin edit. Todo controle existe para VCL, com o prefixo TsgcVCL, e para FireMonkey, com o prefixo TsgcFMX. Cinco variantes data-aware vinculam o seletor de data, o seletor de hora, o spin edit, a avaliação e o controle deslizante de intervalo a um campo de um TDataSet.
O texto é sempre digitado em um edit nativo, nunca em um cursor próprio, de modo que a composição de IME, a área de transferência, o desfazer e os leitores de tela funcionam como em qualquer outro edit do formulário. Todo o resto, a moldura, os botões, o calendário, os chips, as estrelas, os indicadores e o quadrado de cor, é desenhado pelo **sgcUICore**, de modo que um controle tem o mesmo aspecto na VCL, no FireMonkey e na saída PNG e SVG sem janela. Os controles têm temas claro e escuro, uma cor de destaque, suporte completo ao teclado, um nome acessível e escala de DPI por monitor.
Por trás de cada controle há uma classe de modelo independente de framework, como TsgcCalendarModel ou TsgcSpinModel. Os controles VCL e FireMonkey apenas mapeiam a entrada e hospedam o edit e a lista suspensa, de modo que o mesmo modelo também renderiza sem janela, em um servidor ou no Linux64.
O sgcEditors faz parte da suíte visual da eSeGeCe. Ele é desenhado com o sgcUICore, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas, de modo que os editores combinam com os demais produtos da suíte. O sgcEditors exige o sgcUICore, que é instalado junto com ele.



Edições

O sgcEditors é distribuído como Registered, Trial e Community. A Community mantém os nove controles para VCL e FireMonkey e deixa de fora os cinco controles data-aware, o quadrado HSV e a barra alfa do seletor de cor, a lista de preenchimento automático do token edit e o segundo indicador do controle deslizante de intervalo. Veja Edições.

Por onde começar

- Arquitetura: os modelos, os hosts, o edit interno, a lista suspensa e as duas camadas de renderização.
- Controles: os nove controles, suas variantes data-aware e suas classes de modelo em resumo.
- As páginas dos controles: Calendário mensal, Seletor de data, Seletor de hora, Seletor de data e hora, Controle deslizante de intervalo, Seletor de cor, Token edit, Avaliação e Spin edit, os controles data-aware e as classes de modelo, com suas propriedades, métodos, eventos e exemplos.
- Instalação e Plataformas: o instalador, os pacotes, as versões do Delphi e do C++Builder e as plataformas.
- Início rápido VCL, Início rápido FMX, Início rápido C++Builder e Início rápido data-aware: um primeiro formulário em poucos minutos.
- Entrada de texto e IME, Teclado e acessibilidade, Temas, Localização, DPI, Saída sem janela e Edições.
- Solução de problemas: problemas comuns e suas causas.

Arquitetura

O sgcEditors mantém a lógica de edição separada do framework. Cada família de controles tem uma classe de modelo que usa apenas a RTL e o sgcUICore, e os controles VCL e FireMonkey são hosts finos ao redor dela.

As partes de um controle

1. **Modelo.** Um descendente TPersistent de TsgcEditorModel, por exemplo TsgcCalendarModel, TsgcDatePickerModel ou TsgcTokenModel. Guarda o valor e as opções, recebe as teclas e o mouse em pixels lógicos por meio de KeyDown, KeyPress, MouseDown, MouseMove, MouseUp e MouseWheel, e se desenha em qualquer canvas do sgcUICore. Funciona sem janela, de modo que também é o renderizador sem janela.
2. **Host.** O controle VCL descende de TsgcVCLCustomEditor e o controle FireMonkey de TsgcFMXCustomEditor. O host cria o modelo, envia a entrada a ele, pinta suas duas camadas, dispara OnChange e publica as propriedades do modelo. O modelo de um controle é public como Model, e cada controle também tem uma propriedade tipada para ele, como CalendarModel, PickerModel ou SpinModel.
3. **Edit interno.** Os seletores, o spin edit, o seletor de cor e o token edit hospedam um edit nativo sem borda: um TEdit filho na VCL e um TEdit do FMX no FireMonkey. O modelo informa onde vai o texto, e o host posiciona o edit ali. Veja Entrada de texto e IME.
4. **Lista suspensa.** O calendário do seletor de data, a lista de horas do seletor de hora, os dois lado a lado para o seletor de data e hora, e a lista de preenchimento automático do token edit abrem em uma lista suspensa por framework. Na VCL é uma janela de ferramenta WS_POPUP que nunca tira o foco do edit, abre abaixo do controle, passa para cima quando não há espaço e permanece dentro da área de trabalho do monitor. No FireMonkey é um TPopup posicionado abaixo do controle.

Duas camadas

Um modelo desenha duas camadas. Render desenha a camada de conteúdo: a moldura, o texto, os dias, as amostras, os chips e as marcas. RenderOverlay desenha sobre ela o que muda com frequência: o hover, o anel de foco, os indicadores do controle deslizante, o cursor do quadrado de cor, o preenchimento da avaliação e sua prévia de hover. Um arrasto, um hover ou uma mudança de foco redesenha apenas o overlay, de modo que a camada de conteúdo não é rasterizada novamente. Toda alteração de um modelo é informada por OnModelChange com um destes tipos:

Tipo	O host
eckContent	Renderiza a camada de conteúdo novamente.
eckOverlay	Renderiza apenas o overlay.
eckValue	Dispara OnChange e, na VCL, notifica os leitores de tela do novo valor.
eckLayout	Move o edit interno, grava seu texto novamente e renderiza a camada de conteúdo.

Units

Unit	Conteúdo
sgcEditors_Types	As enumerações, TsgcEditorModel, TsgcEditorsPainter (cores do tema, moldura, anel de foco, divisas, setas de spin), códigos de tecla, formatação e interpretação de números, conversão HSV e hexadecimal.
sgcEditors_Date	Números de dia, números de semana ISO 8601 e simples, a grade do mês, TsgcEditorLocale e o formato, o passo e a interpretação de datas.

sgcEditors_Calendar, _Picker, _Slider, _Spin, _Color, _Rating, _Token, _List	Os modelos. sgcEditors_Picker também contém TsgcDateTimeDrop-DownModel, o calendário e a lista de horas da lista suspensa de data e hora.
sgcEditors_VCL, _VCL_Date, _VCL_Value, _VCL_Color, _VCL_Token	A base do host VCL, o edit interno, a lista suspensa e os controles VCL. Somente Windows.
sgcEditors_FMX, _FMX_Date, _FMX_Value, _FMX_Color, _FMX_Token	O mesmo para o FireMonkey, a partir do Delphi XE6.
sgcEditors_Data, _VCL_DB, _FMX_DB	TsgcEditorFieldLink e os controles data-aware. Edição Advanced, essas units ficam vazias na Community.
sgcEditors_Const, _License, _Reg	As constantes de versão e edição, a verificação de avaliação e o registro em tempo de design.

Threads

Os controles são usados a partir da thread principal, como qualquer outro controle VCL ou FireMonkey, e seus eventos são disparados nela.

Desempenho

O grupo de desempenho do QA mede estes cenários com um build release do Delphi 13 em Win64. O orçamento é o limite de cada cenário, e a coluna medida é uma execução na máquina de referência. O calendário mensal também tem um orçamento Win32 de 2,500 µs, e o Linux64 informa E1, E4 e E8 sem limite.

ID	Cenário	Orçamen-to	Medido
E1	Conteúdo do calendário mensal, 280 x 240: título, nomes dos dias, 42 dias, números de semana e uma faixa de intervalo	1,500 µs	395 µs
E2	Somente overlay do calendário: célula em hover e anel de foco	60 µs	24 µs
E3	IsDateEnabled em 10,000 dias com 500 datas desabilitadas, 2 dias da semana desabilitados e o evento atribuído	2 ms	0.3 ms
E4	Controle deslizante de intervalo 300 x 48 com 11 marcas e rótulos, renderização completa; um arrasto de indicador, somente overlay	400 µs; 150 µs	88 µs; 73 µs
E5	Seletor de cor 240 x 300 com 24 amostras, o quadrado, as barras de matiz e alfa, renderização completa; um arrasto no quadrado, somente overlay	1,500 µs; 150 µs	1,018 µs; 114 µs
E6	Token edit com 50 chips, 400 x 120, disposição e renderização; o filtro de preenchimento automático de 10,000 itens a cada tecla	1,500 µs; 3 ms	436 µs; 0.2 ms
E7	Avaliação, 5 estrelas 160 x 32, renderização completa; moldura do spin edit 120 x 28	200 µs; 120 µs	103 µs; 39 µs
E8	100,000 formatações de data mais 100,000 interpretações; 1,000,000 de números de semana ISO	150 ms; 100 ms	23 ms; 27 ms
E9	Um formulário VCL com 30 de cada um dos 9 controles, criado e pinta-do pela primeira vez; uma lista suspensa de seletor aberta até o pri-meiro quadro	350 ms; 16 ms	cerca de 340 ms em uma máqui-

			na ociosa; 5 ms
E10	O formulário E9 no FireMonkey	somente informado	

A maior parte do tempo do E9 é a criação das janelas dos 270 controles e de seus edits internos, não o desenho. Um formulário que mostra centenas de editores de uma vez abre mais rápido quando os controles das páginas ocultas são criados sob demanda.

Veja também

- Controles
- TsgcEditorModel
- Saída sem janela
- Entrada de texto e IME

Controles

O sgcEditors tem nove controles. Cada um existe para VCL e para FireMonkey com as mesmas propriedades e eventos publicados.

Controle	VCL	FireMonkey	Entrada de texto
Calendário mensal	TsgcVCLMonthCalendar	TsgcFMXMonthCalendar	Desenhado. Visões de mês, ano e década, data única ou intervalo, limites, datas e dias da semana desabilitados, números de semana.
Seletor de data	TsgcVCLDatePicker	TsgcFMXDatePicker	Edit nativo mais um calendário suspenso.
Seletor de hora	TsgcVCLTimePicker	TsgcFMXTimePicker	Edit nativo mais uma lista suspensa de horas a cada ListInterval minutos.
Seletor de data e hora	TsgcVCLDateTimePicker	TsgcFMXDateTimePicker	Edit nativo mais o calendário e a lista de horas lado a lado.
Controle deslizante de intervalo	TsgcVCLRangeSlider	TsgcFMXRangeSlider	Desenhado. Um ou dois indicadores, passos, marcas, rótulos e uma dica de valor, horizontal ou vertical.
Seletor de cor	TsgcVCLColorPicker	TsgcFMXColorPicker	Amostras, quadrado HSV e barras de matiz e alfa desenhados, mais um edit hexadecimal nativo.
Token edit	TsgcVCLTokenEdit	TsgcFMXTokenEdit	Chips desenhados, mais um edit nativo para o próximo token e uma lista de preenchimento automático.
Avaliação	TsgcVCLRating	TsgcFMXRating	Desenhado. Estrelas, corações, círculos ou um glifo seu, valores inteiros, meios ou exatos.
Spin edit	TsgcVCLSpinEdit	TsgcFMXSpinEdit	Edit nativo mais botões de subir e descer, unidades, limites e repetição automática.

Controles data-aware

Os controles data-aware descendem do controle da segunda coluna e adicionam DataSource, DataField e um Field público. Fazem parte das edições Registered e Trial. Veja Início rápido data-aware.

VCL	FireMonkey	Controle base	Vincula
TsgcVCLDBDatePicker	TsgcFMXDBDatePicker	Seletor de data	A data do campo, sua parte de hora é mantida.
TsgcVCLDBTimePicker	TsgcFMXDBTimePicker	Seletor de hora	A hora do campo, sua parte de data é mantida.
TsgcVCLDBSpinEdit	TsgcFMXDBSpinEdit	Spin edit	Um campo numérico. Um campo inteiro muda o spin edit para esvInteger.
TsgcVCLDBRating	TsgcFMXDBRating	Avaliação	Um campo numérico.

TsgcVCLDBRangeSlider	TsgcFMXDBRangeSlider	Controle deslizante de intervalo	Value a DataField e, no modo de intervalo, EndValue a EndDataField.

O que todo controle tem

Publicados: Theme, AccentColor, TextStyle, ReadOnly, AccessibleName e OnChange, mais as propriedades e eventos habituais do framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Public: Model, FormatSettings, DroppedDown, DropDown, CloseUp, SaveToPNG e SaveToSVG.

Classes de modelo

Todo modelo descende de TsgcEditorModel.

Modelo	Usado por
TsgcCalendarModel	O calendário mensal e a lista suspensa dos seletores de data.
TsgcDatePickerModel	Os seletores de data, de hora e de data e hora. Kind é edpDate, edpTime ou edpDateTime.
TsgcDateTimeDropDownModel	A lista suspensa do seletor de data e hora: um calendário e uma lista de horas.
TsgcEditorListModel	A lista de horas e a lista de preenchimento automático.
TsgcSliderModel	O controle deslizante de intervalo.
TsgcColorPickerModel	O seletor de cor.
TsgcTokenModel	O token edit.
TsgcRatingModel	A avaliação.
TsgcSpinModel	O spin edit.

Veja também

- Arquitetura
- Início rápido VCL

Instalação

O sgcEditors faz parte da distribuição do sgcUI. Um instalador cobre os produtos do sgcUI e instala apenas os produtos que sua licença inclui.

Instalando com o instalador do sgcUI

1. Execute o instalador do sgcUI e faça login com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcEditors** e os demais produtos que quiser. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que sua licença não inclui ficam esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionado, em ordem de dependência, e instala os pacotes de tempo de design.

O que acompanha o sgcEditors

O **sgcUICore**, a base compartilhada dos produtos visuais do sgcUI, acompanha o sgcEditors. Ele não tem caixa de seleção própria. O instalador o instala automaticamente sempre que o sgcEditors é selecionado.

Pastas instaladas

Cada produto é instalado em uma única pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de todos os produtos instalados. Este é o único caminho de biblioteca do sgcUI.
`Packages`	Os pacotes de todos os produtos instalados, um conjunto por produto e por versão do IDE.
`Lib<V>`	As units compiladas para cada versão do IDE e plataforma, por exemplo `LibD13`.
`Demos\Editors`	As demos do sgcEditors: Gallery, GalleryFMX, Calendar, BookingForm e DataAware.

O instalador adiciona `sgcUI\Source` ao caminho de biblioteca e ao caminho de navegação de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. O sgcUICore fica na mesma pasta `Source`, de modo que não precisa de um caminho próprio.

A paleta de componentes

Os componentes estão na página de paleta **SGC Editors**. Um formulário VCL mostra os controles VCL e um formulário FireMonkey os controles FMX. As edições Registered e Trial registram 28 componentes, os nove controles e os cinco controles data-aware de cada framework. A Community registra os 18 controles sem os data-aware.

Trial e Community

- **Trial**: o instalador trial do sgcUI instala todos os produtos em modo de avaliação, sgcEditors incluído.
- **Community**: o instalador gratuito sgcUI Community inclui o sgcEditors sem seus recursos avançados. Veja Edições.

Compilando os pacotes manualmente

Para compilar os pacotes você mesmo, abra-os para a sua versão do IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo da versão do Delphi, por exemplo `sgcEditorsD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>`	A base compartilhada sgcUICore (runtime).
`sgcEditors<V>` / `dclsgcEditors<V>`	Os componentes do sgcEditors (runtime / tempo de design).

Compile-os em ordem de dependência: `sgcUICore`, depois `sgcEditors`, depois `dclsgcEditors`. Em seguida, instale o pacote de tempo de design `dclsgcEditors` e adicione `sgcUI\Source` ao caminho de biblioteca da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra o projeto `.cbproj` correspondente à sua versão, por exemplo `sgcEditorsC13.cbproj`, e compile-o. Os headers `.hpp` gerados aparecem ao lado do pacote compilado.

Veja também

- Plataformas
- Início rápido VCL

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13. Os pacotes geram os headers .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula.
VCL	Os controles TsgcVCL no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	Os controles TsgcFMX a partir do Delphi XE6 (SGC_EDITORS_FMX), nas plataformas FireMonkey. Usam apenas o FireMonkey e o sgcUICore.
Linux64	As units independentes de framework a partir do Delphi 10.2 (SGC_EDITORS_LINUX): os modelos, a matemática de datas e a localidade, com a saída PNG e SVG sem janela. Não há controles para Linux.
.NET	Uma versão em C# com a mesma API para WinForms, WPF e Avalonia, os nove controles em todos os hosts e a saída PNG e SVG sem janela, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão geral do .NET.

Pacotes

O sgcEditors traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcEditorsD13.dpk para o Delphi 13 e sgcEditorsC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcEditors	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey.	rtl, dbrtl, sgcUICore
sgcEditorsVCL	Runtime, VCL	sgcEditors_VCL, sgcEditors_VCL_Color, sgcEditors_VCL_DB, sgcEditors_VCL_Date, sgcEditors_VCL_Token e sgcEditors_VCL_Value	rtl, vcl, dbrtl, sgcUICore, sgcUICoreVCL, sgcEditors
sgcEditorsFMX	Runtime, FireMonkey, a partir do XE6	sgcEditors_FMX, sgcEditors_FMX_Color, sgcEditors_FMX_DB, sgcEditors_FMX_Date, sgcEditors_FMX_Token e sgcEditors_FMX_Value	rtl, fmx, dbrtl, sgcUICore, sgcUICoreFMX, sgcEditors
dclsgcEditors	Tempo de projeto, VCL	Registra os editores VCL na página **SGC Editors** da paleta, com o editor padrão de lista de strings para DisabledDates e Swatches.	designide, rtl, vcl, sgcEditors, sgcEditorsVCL
dclsgcEditorsFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra os editores FireMonkey na página **SGC Editors** da paleta.	designide, rtl, fmx, sgcEditors, sgcEditorsFMX, dclsgcEditors

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcEditorsFMX exige o dclsgcEditors, portanto instale primeiro o dclsgcEditors.

Versões do IDE e plataformas de destino

Delphi	sgcEditors	sgcEditorsVCL	sgcEditorsFMX	dclsgcEditors, dclsgcEditorsFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcEditors, sgcEditorsVCL	sgcEditorsFMX	dclsgcEditors, dclsgcEditorsFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcEditors)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcEditors)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcEditorsC13.cbproj, sgcEditorsVCLC13.cbproj, sgcEditorsFMXC13.cbproj, dclsgcEditorsC13.cbproj e dclsgcEditorsFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (os editores TsgcFMX) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcEditorsVCL funcionam apenas no Windows.

- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcEditors_FMX, sgcEditors_FMX_Color, sgcEditors_FMX_DB, sgcEditors_FMX_Date, sgcEditors_FMX_Token e sgcEditors_FMX_Value.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcEditors é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa os modelos dos editores (valores, datas, cores, tokens). Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Caixa de texto interna dos editores VCL

Os editores VCL com um campo de texto criam sua caixa de texto interna (uma janela de edição nativa) sob demanda, o que mantém rápida a exibição de formulários com muitos editores. Até lá o editor desenha o texto por conta própria. A caixa de texto interna é criada no primeiro mouse sobre o editor, quando o editor recebe o foco, no primeiro uso do teclado ou da área de transferência, ou quando o código lê o seu handle de janela. Ferramentas de acessibilidade, como leitores de tela e clientes de UI Automation, só veem a caixa de texto interna a partir desse momento. O próprio editor responde a WM_GETTEXT desde o início. Um editor desabilitado ou ReadOnly, ou com PasswordChar, leitura da direita para a esquerda, TextHint ou um Alignment diferente de esquerda, cria a caixa imediatamente, e em tempo de projeto ela é sempre criada.

Restrições de versão

- As units independentes de framework e as units VCL compilam no Delphi 7: sem generics, sem métodos anônimos, sem variáveis inline e sem record helpers.
- TextHint dos controles VCL aparece a partir do Delphi 2009.
- As enumerações têm prefixos de editor (esm, efd, ewn, ecv, eor, etp, ecp, eti, emm, erp, erg, esv, esb), de modo que os headers do C++Builder nunca colidem com os outros produtos do sgcUI.

Veja também

- Edições
- Arquitetura

Início rápido VCL

Instale os pacotes do sgcEditors para a sua versão do Delphi. Os componentes estão na página de paleta **SGC Editors**.

Em tempo de design

1. Coloque um **TsgcVCLDatePicker** em um formulário. Defina MinDate e Date no Object Inspector, ou deixe Date vazio com AllowEmpty definido como True.
2. Coloque um **TsgcVCLSpinEdit**, defina MinValue, MaxValue e Value, e digite um sufixo como guests em Units.
3. Coloque um **TsgcVCLTokenEdit** e dê a ele um TextHint.
4. Atribua um único handler OnChange aos três controles e execute o aplicativo. Digite uma data ou abra o calendário com F4 ou Alt+Down, avance o spin edit com as setas ou a roda do mouse, e digite tokens separados por vírgulas.

No código

Um controle funciona assim que tem um Parent. Defina os limites de um spin edit antes de seu Value, para que o valor não seja limitado aos limites antigos.

```
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;
```

Veja também

- Controles
- Entrada de texto e IME
- Teclado e acessibilidade

Início rápido FMX

Os controles FireMonkey exigem o Delphi XE6 ou posterior. Estão na página de paleta **SGC Editors** de um formulário FireMonkey e têm as mesmas propriedades e eventos que os controles VCL.

Em tempo de design

1. Coloque um **TsgcFMXDatePicker**, um **TsgcFMXRangeSlider** e um **TsgcFMXColorPicker** em um formulário.
2. No controle deslizante, defina RangeMode como True, depois Min, Max, Step e TickInterval, e ative ShowLabels.
3. No seletor de cor, escolha as Parts a mostrar e o SelectedColor.

O que é diferente no FireMonkey

- A lista suspensa é um TPopup abaixo do controle. Quando é fechada por um clique fora dela, as escolhas feitas enquanto estava aberta são desfeitas. Enter, ou a escolha que completa o valor, a fecha e mantém o valor.
- O FireMonkey move o foco com Tab antes de um controle ver a tecla, de modo que o controle deslizante de intervalo e o seletor de cor usam Ctrl com as setas para se mover entre suas partes. Veja Teclado e acessibilidade.
- O texto acessível de um controle vai para o seu Hint, preservando qualquer Hint seu. TabStop, Hint e ShowHint são publicados a partir do Delphi 10.1.
- O edit interno do spin edit solicita o teclado virtual NumbersAndPunctuation nas plataformas móveis.

No 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;
```

Veja também

- Controles
- Plataformas
- DPI

Início rápido C++Builder

Instale os pacotes do sgcEditors para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página de paleta **SGC Editors** exatamente como no Delphi. O instalador adiciona os dois caminhos ao IDE para você. Os headers .hpp e os arquivos .lib gerados do sgcEditors e do sgcUICore estão nessa única pasta.

O que saber em C++

- Inclua o header da unit que declara um controle: `sgcEditors_VCL_Date.hpp` para o calendário e os seletores, `sgcEditors_VCL_Value.hpp` para o controle deslizante, a avaliação e o spin edit, `sgcEditors_VCL_Color.hpp`, `sgcEditors_VCL_Token.hpp` e `sgcEditors_VCL_DB.hpp`. As enumerações estão em `sgcEditors_Types.hpp`.
- Os eventos são closures, de modo que seus handlers são funções membro __fastcall de uma classe. Um parâmetro var como aAccept é uma referência bool.
- Date, MinDate e MaxDate são propriedades TDate. Atribua a elas um TDateTime, por exemplo `Picker->Date = Date() + 1`.
- As classes de modelo têm construtores sem argumentos, como `new TsgcCalendarModel()`, e você mesmo as destrói com delete. Veja Saída sem janela.

No código

Veja também

- Início rápido VCL
- Início rápido data-aware
- Plataformas

Início rápido data-aware

Os controles data-aware vinculam um seletor de data, um seletor de hora, um spin edit, uma avaliação ou um controle deslizante de intervalo a um campo, como o TDBEdit. Fazem parte das edições Registered e Trial, para VCL em sgcEditors_VCL_DB e para FireMonkey em sgcEditors_FMX_DB. O FireMonkey não tem TFieldDataLink, de modo que os dois frameworks compartilham TsgcEditorFieldLink, um TDataLink de sgcEditors_Data.

Em tempo de design

1. Coloque um dataset e um **TDataSource** em um formulário.
2. Coloque um **TsgcVCLDBDatePicker** (ou o gêmeo FMX), defina DataSource como a fonte de dados e escolha o campo em DataField.
3. Para um **TsgcVCLDBRangeSlider** no modo de intervalo, defina DataField para o indicador inferior e EndDataField para o indicador superior.

Como os controles acompanham o campo

- O valor acompanha o registro atual. Um campo nulo mostra um texto vazio nos seletores e no spin edit, nenhum glifo preenchido na avaliação, e o indicador em Min no controle deslizante, com o indicador final em Max. AllowEmpty é True por padrão nos seletores e no spin edit data-aware.
- Uma alteração do usuário coloca o dataset em modo de edição e marca o vínculo como modificado. Definir Value ou Date em código também conta como uma alteração do usuário. Quando o dataset não consegue entrar no modo de edição, o valor do campo retorna.
- Enter e a perda de foco confirmam o texto digitado e gravam o campo. Quando o campo recusa o valor, ele gera uma exceção e o foco permanece no controle, como no TDBEdit. Antes de um post, o dataset grava um valor pendente. Escape restaura o valor do campo.
- O seletor de data mantém a parte de hora de um campo de data e hora, e o seletor de hora mantém sua parte de data. Um campo inteiro muda o spin edit para esvInteger.
- Um controle é somente leitura enquanto seu ReadOnly é True, enquanto não há dataset, enquanto o dataset ou o campo é somente leitura, para um campo calculado e, para o controle deslizante no modo de intervalo, enquanto o campo final não pode ser modificado.

No código

O exemplo cria um TClientDataSet e os cinco controles data-aware da VCL. Adicione MidasLib à cláusula uses do projeto para que o aplicativo não precise de 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;
```

Veja também

- Controles
- Seletor de data DB
- TsgcEditorFieldLink
- Edições
- Solução de problemas

Calendário mensal

Classes: TsgcVCLMonthCalendar (unit sgcEditors_VCL_Date) e TsgcFMXMonthCalendar (unit sgcEditors_FMX_Date)
O calendário mensal é um controle desenhado de forma personalizada, que recebe foco e não tem edit. Mostra um mês com os nomes dos dias e, opcionalmente, os números das semanas, e amplia para os 12 meses de um ano e para os anos de uma década quando o usuário clica no título ou pressiona Ctrl+Up. Seleciona uma data ou, com SelectionMode esmRange, um intervalo de dias. Os dias fora de MinDate e MaxDate, os DisabledDates, os DisabledWeekDays e os dias que OnIsDateDisabled desabilita são desenhados riscados e não podem ser selecionados. O mesmo modelo, TsgcCalendarModel, desenha a lista suspensa dos seletores de data.
A classe VCL descende de TsgcVCLCustomEditor e a classe FireMonkey de TsgcFMXCustomEditor. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Date	TDate	A data selecionada no modo esmSingle, 0 quando nenhuma data está selecionada. O modo de intervalo não a altera.
SelectionMode	TsgcEditorSelectionMode	esmSingle (padrão) seleciona uma data. esmRange seleciona um intervalo: o primeiro clique o inicia, o segundo o termina, e Shift com um clique ou com as setas o estende a partir do primeiro dia.
RangeStart, RangeEnd	TDate	O primeiro e o último dia do intervalo, ambos incluídos.
MaxRangeDays	Integer	O máximo de dias que um intervalo pode conter, incluindo as duas pontas. 0 (padrão) significa sem limite.
MinDate, MaxDate	TDate	O primeiro e o último dia que o usuário pode escolher, comparados como dias inteiros. 0 (padrão) significa sem limite, e limites invertidos são trocados.
DisabledDates	TStrings	Dias que não podem ser escolhidos, uma data ISO (yyyy-mm-dd) por linha.
DisabledWeekDays	TsgcEditorWeekDays	Dias da semana que não podem ser escolhidos, um conjunto de ewdMonday a ewdSunday. Padrão [].
FirstDayOfWeek	TsgcEditorFirstDayOfWeek	efdLocale (padrão) toma o primeiro dia da semana do sistema operacional, segunda-feira quando ele não tem nenhum. efdMonday, efdSaturday e efdSunday o forçam.
ShowWeekNumbers	Boolean	Adiciona uma coluna com o número da semana de cada linha. Padrão False.
WeekNumbering	TsgcEditorWeekNumbering	ewnISO8601 (padrão): a semana 1 contém a primeira quinta-feira do ano. ewnSimple: a semana 1 contém 1 de janeiro.
ShowToday	Boolean	Marca o dia atual. Padrão True.
ShowOtherMonthDays	Boolean	Mostra os dias do mês anterior e do seguinte que completam as seis semanas da grade. Padrão True.

View	TsgcEditorCalendarView	ecvMonth (padrão), ecvYear (os 12 meses de um ano) ou ecvDecade (os anos de uma década).
DisplayMonth	TDate	O primeiro dia do mês mostrado, que é sempre o mês do foco do teclado. Não é armazenado no formulário.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
FormatSettings	TFormatSettings	Public. Os nomes, formatos e separadores que o controle usa. Começa com as configurações da localidade do usuário. Veja Localização.
CalendarModel	TsgcCalendarModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
GoToDate(aDate)		Mostra o mês de aDate, mantido dentro dos limites, e move o foco do teclado para ele. Não seleciona a data.
IsDateEnabled(aDate)	Boolean	True quando a data está dentro dos limites, não está em um dia da semana desabilitado, não está em DisabledDates e não foi desabilitada por OnIsDateDisabled.
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	A data selecionada ou o intervalo mudou.
OnSelect	TNotifyEvent	Uma data é escolhida no modo esmSingle, com o mouse ou com Enter ou Espaço.
OnRangeSelect	TNotifyEvent	Um intervalo está completo: o segundo clique, ou um clique ou uma seta com Shift.

OnIsDate-Disabled	TsgcCalen-darIsDateDi-sabledEvent	(Sender; const aDate: TDateTime; var aDisabled: Boolean). Defina aDisabled como True para desabilitar um dia, como um feriado vindo de um banco de dados. É consultado para os dias que o calendário mostra e para o dia que está sendo selecionado, portanto mantenha-o rápido.
OnView-Change	TNotifyEvent	Visão alterada: o usuário ampliou ou reduziu.
OnDis-play-Month-Change	TNotifyEvent	Outro mês é mostrado.

Teclado

As setas movem o foco em um dia ou uma semana, PageUp e PageDown em um mês e, com Ctrl, em um ano. Ctrl+Up reduz. Enter ou Espaço seleciona o dia focado, ou amplia nas visões de ano e década. Shift com as setas estende um intervalo. Os dias desabilitados são ignorados.

Exemplo: uma estadia de até duas 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;
```

Veja também

- Seletor de data
- Seletor de data e hora
- Localização
- TsgcEditorLocale

Seletor de data

Classes: TsgcVCLDatePicker (unit sgcEditors_VCL_Date) e TsgcFMXDatePicker (unit sgcEditors_FMX_Date)
O seletor de data é um edit nativo para uma data, com um calendário suspenso. O usuário digita a data, avança o dia, o mês ou o ano sob o cursor com Up e Down, ou a escolhe no calendário. Ele compartilha suas classes base, TsgcVCLCustomDatePicker e TsgcFMXCustomDatePicker, e seu modelo, TsgcDatePickerModel, com o seletor de hora e o seletor de data e hora.
A classe VCL descende de TsgcVCLCustomDatePicker e a classe FireMonkey de TsgcFMXCustomDatePicker. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Date	TDate	A data. Leia IsEmpty antes quando AllowEmpty é True.
DisplayFormat	string	O formato do texto, como dd/mm/yyyy ou ddd d mmm yyyy. Vazio (padrão) usa o formato de data curta de FormatSettings. Veja Entrada de texto e IME.
AllowEmpty	Boolean	Permite que o usuário apague o texto, o que esvazia o valor. Padrão False, e então um texto vazio devolve o último valor.
IsEmpty	Boolean	Public, somente leitura. True enquanto o valor está vazio.
TextHint	string	A dica cinza de um edit vazio. A VCL a mostra a partir do Delphi 2009, e o FireMonkey a mostra como o TextPrompt do edit.
DroppedDown	Boolean	Public. True enquanto a lista suspensa está aberta. Atribuir um valor a abre ou fecha.
MinDate, MaxDate	TDate	O primeiro e o último dia que o usuário pode escolher, comparados como dias inteiros. 0 (padrão) significa sem limite, e limites invertidos são trocados.
DisabledDates	TStrings	Dias que não podem ser escolhidos, uma data ISO (yyyy-mm-dd) por linha.
DisabledWeekDays	TsgcEditorWeekDays	Dias da semana que não podem ser escolhidos, um conjunto de ewdMonday a ewdSunday. Padrão [].
FirstDayOfWeek, ShowWeekNumbers		Como no calendário mensal, para o calendário suspenso.
DropDownCalendar	TsgcCalendarModel	Public, somente leitura. O modelo do calendário suspenso.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.

TextStyle	TsgcUI-TextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
FormatSettings	TFormat-Settings	Public. Os nomes, formatos e separadores que o controle usa. Começa com as configurações da localidade do usuário. Veja Localização.
PickerModel	TsgcDate-PickerMo-del	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retor-na	Descrição
Clear		Esvazia o valor quando AllowEmpty é True. Caso contrário, não faz na-da.
DropDown		Abre a lista suspensa, como o botão, F4 e Alt+Down.
CloseUp(aAccept)		Fecha a lista suspensa. aAccept True confirma o texto, False restaura o valor que tinha ao abrir.
IsDateEnabled(aDate)	Boole-an	True quando a data está dentro dos limites e não está desabilitada no calendário suspenso. Uma confirmação recusa as datas para as quais é False.
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multipli-ca o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyE-vent	Um novo valor: digitado e confirmado, escolhido na lista suspensa ou alterado com Up e Down.
OnDrop-Down, On-CloseUp	TNotifyE-vent	A lista suspensa abre ou fecha.
OnParseEr-ror	TsgcDate-PickerPar-seErrorE-vent	(Sender; const aText: string; var aValue: TDateTime; var aAccept: Boolean). O texto confirmado não pode ser interpretado. aValue chega como o valor atual e aAccept como False. Defina aValue e aAccept como True para aceitar outro valor, que então é mantido dentro dos limites. Caso contrário, o último valor válido retorna.

OnIsDate-Disabled	TsgcCalen-darIsDate-DisabledE-vent	Como no calendário mensal: desabilita um dia na lista suspensa e na confir-mação.

Digitação e lista suspensa

- O texto é digitado em um edit nativo, de modo que IME, a área de transferência e o desfazer funcionam. A interpretação é tolerante e segue os nomes e separadores de FormatSettings. Veja Entrada de texto e IME.
- Up e Down confirmam o texto digitado e depois avançam o campo sob o cursor, que dá a volta dentro do seu intervalo, e selecionam o campo novamente. Os limites são aplicados após o passo.
- Enter, a perda de foco e uma escolha na lista suspensa confirmam o texto. Um valor fora dos limites é movi-do para dentro deles, e uma data desabilitada é recusada.
- F4, Alt+Down e o botão abrem a lista suspensa no valor atual. Enquanto está aberta, ela acompanha o texto digitado, Up, Down, PageUp e PageDown vão até ela, e Escape a fecha e restaura o valor e o texto que ti-nha ao abrir.

Exemplo: uma data de vencimento que entende "hoje"

```
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;
```

Veja também

- Calendário mensal
- Seletor de data e hora
- Seletor de data DB
- Entrada de texto e IME

Seletor de hora

Classes: TsgcVCLTimePicker (unit sgcEditors_VCL_Date) e TsgcFMXTimePicker (unit sgcEditors_FMX_Date)
O seletor de hora é um edit nativo para uma hora do dia, com uma lista suspensa de horas. O usuário digita a hora, avança a hora, o minuto, o segundo ou AM e PM sob o cursor com Up e Down, ou escolhe uma hora na lista. A lista começa em 00:00 e contém uma hora a cada ListInterval minutos entre MinTime e MaxTime, escrita com os campos de hora de DisplayFormat.
A classe VCL descende de TsgcVCLCustomDatePicker e a classe FireMonkey de TsgcFMXCustomDatePicker. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Time	TTime	A hora do dia. O seletor de hora começa em 00:00.
DisplayFormat	string	O formato do texto, como hh:nn ou h:nn ampm. Um formato com ss mostra os segundos. Vazio (padrão) usa o formato de hora curta de FormatSettings.
MinTime, MaxTime	TTime	A hora mais cedo e a mais tarde que o usuário pode definir. 0 (padrão) significa sem limite nesse lado.
ListInterval	Integer	Os minutos entre duas horas da lista suspensa. Padrão 30. 0 remove a lista suspensa e seu botão.
AllowEmpty	Boolean	Permite que o usuário apague o texto, o que esvazia o valor. Padrão False, e então um texto vazio devolve o último valor.
IsEmpty	Boolean	Public, somente leitura. True enquanto o valor está vazio.
TextHint	string	A dica cinza de um edit vazio. A VCL a mostra a partir do Delphi 2009, e o FireMonkey a mostra como o TextPrompt do edit.
DroppedDown	Boolean	Public. True enquanto a lista suspensa está aberta. Atribuir um valor a abre ou fecha.
DropDownTimeList	TsgcEditorListModel	Public, somente leitura. O modelo da lista suspensa.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.

FormatSettings	TFormatSettings	Public. Os nomes, formatos e separadores que o controle usa. Começa com as configurações da localidade do usuário. Veja Localização.
PickerModel	TsgcDatePickerModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
Clear		Esvazia o valor quando AllowEmpty é True. Caso contrário, não faz nada.
DropDown		Abre a lista suspensa, como o botão, F4 e Alt+Down.
CloseUp(aAccept)		Fecha a lista suspensa. aAccept True confirma o texto, False restaura o valor que tinha ao abrir.
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Um novo valor: digitado e confirmado, escolhido na lista suspensa ou alterado com Up e Down.
OnDropDown, OnCloseUp	TNotifyEvent	A lista suspensa abre ou fecha.
OnParseError	TsgcDatePickerParseErrorEvent	(Sender; const aText: string; var aValue: TDateTime; var aAccept: Boolean). O texto confirmado não pode ser interpretado. aValue chega como o valor atual e aAccept como False. Defina aValue e aAccept como True para aceitar outro valor, que então é mantido dentro dos limites. Caso contrário, o último valor válido retorna.

Digitação e lista suspensa

- O texto é digitado em um edit nativo, de modo que IME, a área de transferência e o desfazer funcionam. A interpretação é tolerante e segue os nomes e separadores de FormatSettings. Veja Entrada de texto e IME.
- Up e Down confirmam o texto digitado e depois avançam o campo sob o cursor, que dá a volta dentro do seu intervalo, e selecionam o campo novamente. Os limites são aplicados após o passo.
- Enter, a perda de foco e uma escolha na lista suspensa confirmam o texto. Um valor fora dos limites é movido para dentro deles, e uma data desabilitada é recusada.
- F4, Alt+Down e o botão abrem a lista suspensa no valor atual. Enquanto está aberta, ela acompanha o texto digitado, Up, Down, PageUp e PageDown vão até ela, e Escape a fecha e restaura o valor e o texto que tinha ao abrir.

Exemplo: uma reunião em horário comercial

```
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;
```

Veja também

- Seletor de data
- Seletor de data e hora
- Seletor de hora DB

Seletor de data e hora

Classes: TsgcVCLDateTimePicker (unit sgcEditors_VCL_Date) e TsgcFMXDateTimePicker (unit sgcEditors_FMX_Date)
O seletor de data e hora edita uma data e uma hora em um único texto. Sua lista suspensa mostra o calendário e a lista de horas lado a lado. Uma escolha no calendário mantém a lista aberta, para que o usuário escolha a hora em seguida, e uma escolha na lista de horas a fecha. MinDate, MaxDate e os dias desabilitados limitam a data, e MinTime e MaxTime limitam a hora do dia.
A classe VCL descende de TsgcVCLCustomDatePicker e a classe FireMonkey de TsgcFMXCustomDatePicker. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
DateTime	TDateTime	A data e a hora.
DisplayFormat	string	O formato do texto, como dd/mm/yyyy hh:nn. Vazio (padrão) usa a data curta e a hora curta de FormatSettings, separadas por um espaço.
MinDate, MaxDate	TDate	O primeiro e o último dia que o usuário pode escolher, comparados como dias inteiros. 0 (padrão) significa sem limite, e limites invertidos são trocados.
DisabledDates	TStrings	Dias que não podem ser escolhidos, uma data ISO (yyyy-mm-dd) por linha.
DisabledWeekDays	TsgcEditorWeekDays	Dias da semana que não podem ser escolhidos, um conjunto de ewdMonday a ewdSunday. Padrão [].
MinTime, MaxTime	TTime	A hora do dia mais cedo e a mais tarde. 0 (padrão) significa sem limite nesse lado.
ListInterval	Integer	Os minutos entre duas horas da lista de horas. Padrão 30.
FirstDayOfWeek, ShowWeekNumbers		Como no calendário mensal, para o calendário suspenso.
AllowEmpty	Boolean	Permite que o usuário apague o texto, o que esvazia o valor. Padrão False, e então um texto vazio devolve o último valor.
IsEmpty	Boolean	Public, somente leitura. True enquanto o valor está vazio.
TextHint	string	A dica cinza de um edit vazio. A VCL a mostra a partir do Delphi 2009, e o FireMonkey a mostra como o TextPrompt do edit.
DroppedDown	Boolean	Public. True enquanto a lista suspensa está aberta. Atribuir um valor a abre ou fecha.
DropDownCalendar, DropDownTimeList	TsgcCalendarModel, TsgcEditorListModel	Public, somente leitura. Os modelos das duas partes da lista suspensa.

Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
FormatSettings	TFormatSettings	Public. Os nomes, formatos e separadores que o controle usa. Começa com as configurações da localidade do usuário. Veja Localização.
PickerModel	TsgcDatePickerModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
Clear		Esvazia o valor quando AllowEmpty é True. Caso contrário, não faz nada.
DropDown		Abre a lista suspensa, como o botão, F4 e Alt+Down.
CloseUp(aAccept)		Fecha a lista suspensa. aAccept True confirma o texto, False restaura o valor que tinha ao abrir.
IsDateEnabled(aDate)	Boolean	True quando a data está dentro dos limites e não está desabilitada no calendário suspenso.
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Um novo valor: digitado e confirmado, escolhido na lista suspensa ou alterado com Up e Down.
OnDropDown, OnCloseUp	TNotifyEvent	A lista suspensa abre ou fecha.
OnParseError	TsgcDatePickerPar-	(Sender; const aText: string; var aValue: TDateTime; var aAccept: Boolean). O texto confirmado não pode ser interpretado. aValue chega como o valor atual e aAccept como False. Defina aValue e aAccept como True para aceitar outro

	seErrorEvent	valor, que então é mantido dentro dos limites. Caso contrário, o último valor válido retorna.
OnIsDateDisabled	TsgcCalendarIsDateDisabledEvent	Como no calendário mensal.

Digitação e lista suspensa

- O texto é digitado em um edit nativo, de modo que IME, a área de transferência e o desfazer funcionam. A interpretação é tolerante e segue os nomes e separadores de FormatSettings. Veja Entrada de texto e IME.
- Up e Down confirmam o texto digitado e depois avançam o campo sob o cursor, que dá a volta dentro do seu intervalo, e selecionam o campo novamente. Os limites são aplicados após o passo.
- Enter, a perda de foco e uma escolha na lista suspensa confirmam o texto. Um valor fora dos limites é movido para dentro deles, e uma data desabilitada é recusada.
- F4, Alt+Down e o botão abrem a lista suspensa no valor atual. Enquanto está aberta, ela acompanha o texto digitado, Up, Down, PageUp e PageDown vão até ela, e Escape a fecha e restaura o valor e o texto que tinha ao abrir.

Na lista suspensa do seletor de data e hora, Up, Down, PageUp e PageDown vão para o calendário quando o cursor está em um campo de data e para a lista de horas quando está em um campo de hora.

Exemplo: um horário 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;
```

Veja também

- Seletor de data
- Seletor de hora
- Calendário mensal

Controle deslizante de intervalo

Classes: TsgcVCLRangeSlider (unit sgcEditors_VCL_Value) e TsgcFMXRangeSlider (unit sgcEditors_FMX_Value)
O controle deslizante de intervalo é um controle desenhado de forma personalizada, que recebe foco e escolhe um número entre Min e Max com um indicador, ou um intervalo com dois indicadores em RangeMode. É horizontal, com Min à esquerda, ou vertical, com Min embaixo. Pode desenhar marcas, rótulos e uma dica com o valor sobre o indicador arrastado. Um clique no trilho move o indicador mais próximo até lá e o arrasta, e os indicadores nunca se cruzam.
RangeMode exige a edição Registered ou Trial. Na Community a propriedade é mantida, um indicador é desenhado e editado, e EndValue é mantido, mas não é mostrado. Veja Edições.
A classe VCL descende de TsgcVCLCustomEditor e a classe FireMonkey de TsgcFMXCustomEditor. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Min, Max	Double	Os extremos da escala. Padrão 0 e 100.
Value	Double	O valor, ou a extremidade inferior do intervalo em RangeMode. Padrão 0.
EndValue	Double	A extremidade superior do intervalo em RangeMode. Padrão 100.
RangeMode	Boolean	Dois indicadores, Value e EndValue. Padrão False. Edições Registered e Trial.
MinRange	Double	A menor distância entre os dois indicadores. Padrão 0.
Step	Double	Todo valor se ajusta a Min mais um múltiplo de Step, e as setas movem um Step. Padrão 1. 0 é contínuo e, então, as setas movem 1% da escala.
LargeStep	Double	O passo de PageUp e PageDown. Padrão 10. 0 é um décimo da escala.
Orientation	TsgcEditorOrientation	eorHorizontal (padrão) ou eorVertical.
TickInterval	Double	A distância entre duas marcas. 0 (padrão) não desenha marcas.
TickPlacement	TsgcEditorTickPlacement	etpBottomRight (padrão), etpTopLeft, etpBoth ou etpNone.
ShowLabels	Boolean	Escreve o valor em cada marca, ou em Min e Max quando não há marcas. Os rótulos que se sobreporiam são omitidos. Padrão False.
LabelFormat	string	Uma máscara FormatFloat para os rótulos e a dica de valor. Vazio (padrão) escreve tantos decimais quanto Step precisa.
ShowValueTip	Boolean	Mostra o valor sobre o indicador enquanto ele é arrastado. Padrão True.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.

TextStyle	TsgcUI-TextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
Accessible-Name	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
FormatSettings	TFormatSettings	Public. Os nomes, formatos e separadores que o controle usa. Começa com as configurações da localidade do usuário. Veja Localização.
SliderModel	TsgcSliderModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Value ou EndValue mudou. Dispara durante um arrasto, para cada novo valor.
OnChangeComplete	TNotifyEvent	O usuário terminou uma alteração: soltar o mouse, soltar a tecla ou um passo da roda. Use-o para trabalho que não deve rodar durante um arrasto, como uma consulta.
OnFormatLabel	TsgcSliderFormatLabelEvent	(Sender; aValue: Double; var aText: string). Escreve o texto de um rótulo e da dica de valor, como uma moeda ou uma unidade.

Teclado

As setas movem o indicador ativo um Step, PageUp e PageDown um LargeStep, e Home e End vão para os extremos. Em RangeMode, Tab vai do indicador inferior ao superior e Shift+Tab volta, e então o Tab seguinte sai do controle. Ctrl+Left ou Ctrl+Down ativa o indicador inferior, e Ctrl+Right ou Ctrl+Up o superior. No FireMonkey, Tab sempre vai ao controle seguinte, então as teclas Ctrl alternam os indicadores.

Exemplo: um filtro de preço

```
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;
```

Veja também

- Controle deslizante de intervalo DB
- Spin edit
- Teclado e acessibilidade

Seletor de cor

Classes: TsgcVCLColorPicker (unit sgcEditors_VCL_Color) e TsgcFMXColorPicker (unit sgcEditors_FMX_Color)
O seletor de cor mostra, de cima para baixo, uma grade de amostras, um quadrado de saturação e valor, uma barra de matiz, uma barra alfa e uma linha com uma prévia e um edit nativo para o texto hexadecimal. Parts escolhe quais delas são mostradas. O edit hexadecimal aceita textos #RGB, #RGBA, #RRGGBB e #RRGGBBAA. O seletor guarda o matiz separado da cor, de modo que um cinza mantém seu matiz e o quadrado não salta.
O quadrado HSV e a barra alfa exigem a edição Registered ou Trial. Na Community eles não são desenhados, o alfa é sempre $FF e o texto hexadecimal é #RRGGBB. Veja Edições.
A classe VCL descende de TsgcVCLCustomEditor e a classe FireMonkey de TsgcFMXCustomEditor. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
SelectedColor	TsgcUIColor	A cor como ARGB, por exemplo $FF2A78D6. Tem o layout do TAlphaColor do FireMonkey.
SelectedVCLColor	TColor	Somente VCL, public. A cor como um TColor opaco, para leitura e escrita.
Parts	TsgcEditorColorParts	As partes mostradas: ecpSwatches, ecpHSV, ecpAlpha e ecpHex. Padrão: as quatro.
Swatches	TStrings	As cores das amostras, uma cor hexadecimal como #1E88E5 por linha. Padrão: 24 cores das paletas do sgcUICore e tons de cinza.
SwatchSize	Double	O tamanho de uma amostra em pixels lógicos. Padrão 20.
SwatchColumns	Integer	As amostras em uma linha. Padrão 8.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
ColorModel	TsgcColorPickerModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição

SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	A cor mudou. Dispara durante um arrasto no quadrado ou nas barras, para cada nova cor.
OnSelect	TNotifyEvent	Uma cor é escolhida: um clique em uma amostra, o soltar do mouse após um arrasto, Enter, ou um texto hexadecimal confirmado com Enter ou pela perda de foco.

Teclado

- Na VCL o controle tem dois pontos de foco. Tab percorre as partes, depois entra no edit hexadecimal, e o Tab seguinte sai. Em somente leitura, o controle é um único ponto de foco.
- No FireMonkey o controle é uma única parada de tabulação com duas paradas internas: Ctrl+setas alternam entre as partes e o edit hexadecimal.
- Nas amostras, as setas, Home e End movem o foco e Enter ou Espaço escolhem a amostra. No quadrado, Left e Right alteram a saturação e Up e Down o valor em 1%. Nas barras, as setas movem em um grau ou 1%. Shift anda dez vezes mais.
- Um texto hexadecimal que não é uma cor mostra a cor novamente, e Escape a restaura.

Exemplo: cores da marca para um rótulo

```
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;
```

Exemplo: uma cor de preenchimento no 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;
```

Veja também

- Token edit
- Temas
- Edições

Token edit

Classes: TsgcVCLTokenEdit (unit sgcEditors_VCL_Token) e TsgcFMXTokenEdit (unit sgcEditors_FMX_Token)
O token edit contém uma lista de tokens, como tags ou endereços de e-mail, desenhados como chips com um botão de remover. Os chips quebram da esquerda para a direita, e um edit nativo após o último chip recebe o token seguinte, de modo que IME, a área de transferência e o desfazer funcionam. Um delimitador, Enter, Tab e a perda de foco transformam o texto digitado em um chip, e colar a, b, c cria três chips. Cada token é aparado, verificado por OnValidateToken, verificado quanto a duplicatas e contado em relação a MaxTokens. Quando os chips não cabem, o controle rola na vertical.
A lista de preenchimento automático exige a edição Registered ou Trial. Na Community as propriedades são mantidas e a lista nunca abre. Veja Edições.
A classe VCL descende de TsgcVCLCustomEditor e a classe FireMonkey de TsgcFMXCustomEditor. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Tokens	TStrings	Os tokens, em ordem. Alterar a lista altera os chips.
Delimiters	string	Os caracteres que encerram um token. Padrão ,; e as quebras de linha sempre dividem. Vazio deixa apenas Enter e as quebras de linha.
AllowDuplicates	Boolean	Adiciona um token que já está na lista. Padrão False.
CaseSensitive	Boolean	Compara os tokens diferenciando maiúsculas de minúsculas ao procurar duplicatas. Padrão False.
MaxTokens	Integer	O máximo de tokens que a lista aceita. 0 (padrão) significa sem limite. Um texto acima do limite permanece no edit.
InvalidTokens	TsgcEditorInvalidTokens	etiReject (padrão) mantém no edit um texto recusado. etiMark o adiciona como um chip marcado; TokenModel.IsTokenInvalid(aIndex) informa qual.
TextHint	string	A dica cinza do edit vazio.
AutoCompleteItems	TStrings	Os candidatos da lista de preenchimento automático. Edições Registered e Trial.
AutoCompleteMatch	TsgcEditorMatchMode	emmStartsWith (padrão) ou emmContains.
AutoCompleteMinChars	Integer	Os caracteres que o usuário digita antes de a lista abrir. Padrão 1.
AutoCompleteMaxItems	Integer	O máximo de linhas que a lista mostra. 0 (padrão) significa sem limite.
DroppedDown	Boolean	Public. True enquanto a lista de preenchimento automático está aberta.

Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
TokenModel	TsgcTokenModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
AddToken(aText)	Boolean	Adiciona um token sem dividi-lo, passando por apara, OnValidateToken, verificação de duplicatas e MaxTokens. True quando foi adicionado.
RemoveToken(aIndex)	Boolean	Remove o token em aIndex. True quando foi removido.
Clear		Remove todos os tokens e o texto digitado.
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	A lista de tokens mudou.
OnValidateToken	TsgcTokenValidateEvent	(Sender; var aText: string; var aValid: Boolean). aText chega aparado e pode ser alterado, por exemplo para normalizá-lo. Defina aValid como False para recusar o token, ou para marcá-lo com etiMark.
OnTokenAdded, OnTokenRemoved	TsgcTokenEvent	(Sender; aIndex: Integer; const aText: string). Um chip foi adicionado ou removido.
OnAutoComplete	TsgcTokenAutoCompleteEvent	(Sender; const aText: string; aItems: TStrings). aItems são os candidatos da lista. Altere-os conforme o texto digitado, por exemplo a partir de um banco de dados, e a lista então os filtra. Edições Registered e Trial.

Teclado

Backspace ou Left em um edit vazio focaliza o último chip, e um segundo Backspace o remove. Left e Right percorrem os chips, Right no último volta ao edit, Delete remove o chip focalizado, e Escape ou digitar volta ao edit. Enquanto a lista de preenchimento automático está aberta, Up, Down, PageUp e PageDown se movem nela, Enter escolhe a linha destacada e Escape a fecha.

Exemplo: destinatários de e-mail

```
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;
```

Veja também

- Entrada de texto e IME
- Teclado e acessibilidade
- Edições

Avaliação

Classes: TsgcVCLRating (unit sgcEditors_VCL_Value) e TsgcFMXRating (unit sgcEditors_FMX_Value)
A avaliação é uma fileira de glifos desenhada de forma personalizada, que recebe foco: estrelas, corações, círculos ou um glifo seu. Um clique escolhe o valor sob o mouse, e passar o mouse mostra uma prévia de um valor sem alterá-lo. Precision arredonda o valor para glifos inteiros ou metades, ou o mantém exato para uma média somente leitura como 3.7, desenhada com um glifo parcialmente preenchido.
A classe VCL descende de TsgcVCLCustomEditor e a classe FireMonkey de TsgcFMXCustomEditor. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Value	Double	O valor, de 0 a Count, arredondado por Precision.
Count	Integer	O número de glifos, de 1 a 20. Padrão 5.
Precision	TsgcEditorRatingPrecision	erpWhole (padrão) arredonda para glifos inteiros, erpHalf para metades, e erpExact não arredonda. Defina Count e Precision antes de Value.
Glyph	TsgcEditorRatingGlyph	ergStar (padrão), ergHeart, ergCircle ou ergCustom, que desenha por meio de OnDrawGlyph.
GlyphSize	Double	O tamanho de um glifo em pixels lógicos. 0 (padrão) é o maior tamanho que cabe no controle.
Spacing	Double	Os pixels lógicos entre dois glifos. Padrão 4.
FillColor	TsgcUIColor	A cor dos glifos preenchidos. 0 (padrão) é o âmbar da paleta do tema.
EmptyColor	TsgcUIColor	A cor dos glifos vazios. 0 (padrão) é a cor de borda do tema.
AllowClear	Boolean	Um clique no valor atual o limpa para 0, e Home vai para 0. Padrão False e, então, o menor valor que o usuário pode escolher é um passo.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.
AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
RatingModel	TsgcRatingModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Valor alterado.
OnHover	TsgcRatingHoverEvent	(Sender; aValue: Double). O valor sob o mouse durante a prévia, e -1 quando a prévia termina.
OnDrawGlyph	TsgcRatingDrawGlyphEvent	(Sender; aCanvas: TsgcUICanvas; const aRect: TsgcUIRect; aFill: Double). Desenha um glifo de ergCustom em aRect, em pixels de dispositivo. É chamado com aFill 0 para o glifo vazio e, depois, novamente com a parte preenchida, de 0 a 1 a partir da esquerda.

Teclado

Right e Up somam um passo da precisão, Left e Down subtraem um. Home vai ao menor valor, End a Count, e as teclas de dígitos definem um valor inteiro.

Exemplo: uma nota de zero a dez com um glifo seu

```
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;
```

Veja também

- Avaliação DB
- Controle deslizante de intervalo
- Temas

Spin edit

Classes: TsgcVCLSpinEdit (unit sgcEditors_VCL_Value) e TsgcFMXSpinEdit (unit sgcEditors_FMX_Value)
O spin edit é um edit nativo para um número, com botões de subir e descer. O valor é um Double: esvInteger o arredonda para números inteiros e esvFloat para Decimals. Units é um sufixo discreto, como kg ou %, desenhado após o texto e que nunca faz parte dele. Um botão mantido pressionado repete, mais rápido quanto mais tempo é mantido. A interpretação é tolerante: espaços e o separador de milhar são ignorados, e tanto . quanto , são aceitos como separador decimal.
A classe VCL descende de TsgcVCLCustomEditor e a classe FireMonkey de TsgcFMXCustomEditor. Ambas publicam as propriedades e os eventos da tabela abaixo, além dos habituais do seu framework, como Align, Anchors, Enabled, Hint, TabOrder, TabStop, OnEnter, OnExit e os eventos de teclado e de mouse. Veja Controles.

Propriedades

Propriedade	Tipo	Descrição
Value	Double	O valor. Defina ValueType, MinValue e MaxValue antes, ou Value é mantido dentro dos limites antigos.
MinValue, MaxValue	Double	Os limites. Valores iguais, como o padrão 0 e 0, significam sem limite.
Increment	Double	O passo dos botões, de Up, Down e da roda. Padrão 1.
LargeIncrement	Double	O passo de PageUp e PageDown. Padrão 10.
ValueType	TsgcEditorSpinValueType	esvInteger (padrão) ou esvFloat.
Decimals	Integer	As casas decimais de um valor esvFloat. Padrão 2.
DisplayFormat	string	Uma máscara FormatFloat para o texto, como #,##0.00. Vazio (padrão) escreve o valor com Decimals.
Units	string	Um sufixo discreto após o texto, que não faz parte do valor.
Buttons	TsgcEditorSpinButtons	esbRight (padrão) empilha subir e descer à direita, esbSides coloca menos à esquerda e mais à direita, esbNone não tem botões.
Wrap	Boolean	Um passo além de um limite primeiro para no limite, e o passo seguinte vai para a outra extremidade. Padrão False.
AllowEmpty	Boolean	Permite que o usuário apague o texto, o que esvazia o valor. Padrão False.
IsEmpty	Boolean	Public, somente leitura. True enquanto o valor está vazio.
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	A cor de destaque ARGB da seleção, do preenchimento e do anel de foco. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto. Um Color 0 segue o tema.
ReadOnly	Boolean	O valor é mostrado, mas o usuário não pode alterá-lo. Padrão False.

AccessibleName	string	O nome que um leitor de tela anuncia antes do valor. Veja Teclado e acessibilidade.
FormatSettings	TFormatSettings	Public. Os nomes, formatos e separadores que o controle usa. Começa com as configurações da localidade do usuário. Veja Localização.
SpinModel	TsgcSpinModel	Public, somente leitura. O modelo do controle. Model retorna o mesmo objeto como um TsgcEditorModel.

Métodos

Método	Retorna	Descrição
Clear		Esvazia o valor quando AllowEmpty é True. Caso contrário, não faz nada.
SaveToPNG(aStream ou aFileName, aScale = 1)		Desenha o controle no seu tamanho atual em um PNG. aScale multiplica o tamanho. Veja Saída sem janela.
SaveToSVG(aStream ou aFileName, aScale = 1)		O mesmo em um SVG.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	Um novo valor: confirmado, avançado por passos ou definido em código.
OnParseError	TsgcSpinParseErrorEvent	(Sender; const aText: string; var aValue: Double; var aAccept: Boolean). O texto confirmado não é um número. Defina aValue e aAccept como True para aceitar um valor. Caso contrário, o último valor retorna.
OnValidate	TsgcSpinValidateEvent	(Sender; var aValue: Double; var aAccept: Boolean). Chamado antes de um novo valor ser definido. aAccept chega como True. Altere aValue para ajustá-lo, ou defina aAccept como False para manter o valor antigo.

Teclado e mouse

Up e Down avançam um Increment, PageUp e PageDown um LargeIncrement, e a roda um Increment por clique. Cada um deles confirma antes o texto digitado. Enter e a perda de foco confirmam o texto, e Escape o restaura. Um botão mantido pressionado repete: um Increment por repetição, depois 2 a partir da 10ª repetição, 5 a partir da 20ª e 10 a partir da 30ª.

Exemplo: um ângulo em meios graus

```
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;
```

Veja também

- Spin edit DB
- Controle deslizante de intervalo
- Localização

Seletor de data DB

Classes: TsgcVCLDBDatePicker (unit sgcEditors_VCL_DB) e TsgcFMXDBDatePicker (unit sgcEditors_FMX_DB)
O seletor de data DB vincula a data de um seletor de data a um campo de data ou de data e hora. Descende de TsgcVCLCustomDatePicker e TsgcFMXCustomDatePicker e publica as propriedades do seletor de data, exceto Date: o valor é o campo. As demais propriedades, métodos e eventos publicados são os do seletor de data.
Os controles data-aware fazem parte das edições Registered e Trial. Suas units ficam vazias na Community. Eles se vinculam por meio de um TsgcEditorFieldLink, que usa apenas a unit DB, de modo que os controles FireMonkey não precisam de LiveBindings.

Propriedades

Propriedade	Tipo	Descrição
DataSource	TDataSource	A fonte de dados do dataset.
DataField	string	O nome do campo ao qual o valor está vinculado.
AllowEmpty	Boolean	Padrão True, de modo que um campo nulo mostra um texto vazio e apagar o texto grava nulo.
DisplayFormat, MinDate, MaxDate, DisabledDates, DisabledWeekDays, FirstDayOfWeek, ShowWeekNumbers, TextHint		Como no seletor de data.
ReadOnly	Boolean	True mantém o controle somente leitura. Ele também é somente leitura enquanto o campo não pode ser modificado: sem dataset, um dataset ou campo somente leitura, ou um campo calculado. Padrão False.
Field	TField	Public, somente leitura. O campo vinculado, nil enquanto o vínculo não está ativo.

Métodos

Método	Retorna	Descrição
SaveToPNG, SaveToSVG		Como no seletor de data. Os demais métodos do controle base também se aplicam.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O valor mudou, pelo usuário, em código ou porque o registro se moveu.
OnDropDown, OnCloseUp, OnParseError, OnIsDateDisabled		Como no seletor de data.

Como o campo é editado

- O valor acompanha o registro. Um campo nulo mostra um texto vazio.
- Uma alteração do usuário coloca o dataset em modo de edição e marca o valor como modificado. Definir o valor em código também é uma alteração. Quando o dataset não consegue entrar no modo de edição, o valor do campo retorna.
- Enter e a perda de foco gravam o campo, e o dataset grava um valor pendente antes de confirmar. Um campo que recusa o valor gera uma exceção e o foco permanece no controle, como no TDBEdit.
- Escape restaura o valor do campo.
- O seletor de data mantém a parte de hora de um TDateTimeField e altera apenas a data.

Exemplo: uma data de nascimento

```
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;
```

Veja também

- Seletor de data
- Seletor de hora DB
- Início rápido data-aware
- TsgcEditorFieldLink

Seletor de hora DB

Classes: TsgcVCLDBTimePicker (unit sgcEditors_VCL_DB) e TsgcFMXDBTimePicker (unit sgcEditors_FMX_DB)
O seletor de hora DB vincula a hora de um seletor de hora a um campo de hora ou de data e hora. Descende de TsgcVCLCustomDatePicker e TsgcFMXCustomDatePicker e publica as propriedades do seletor de hora, exceto Time: o valor é o campo. As demais propriedades, métodos e eventos publicados são os do seletor de hora.
Os controles data-aware fazem parte das edições Registered e Trial. Suas units ficam vazias na Community. Eles se vinculam por meio de um TsgcEditorFieldLink, que usa apenas a unit DB, de modo que os controles FireMonkey não precisam de LiveBindings.

Propriedades

Propriedade	Tipo	Descrição
DataSource	TDataSource	A fonte de dados do dataset.
DataField	string	O nome do campo ao qual o valor está vinculado.
AllowEmpty	Boolean	Padrão True, de modo que um campo nulo mostra um texto vazio e apagar o texto grava nulo.
DisplayFormat, MinTime, MaxTime, ListInterval, TextHint		Como no seletor de hora.
ReadOnly	Boolean	True mantém o controle somente leitura. Ele também é somente leitura enquanto o campo não pode ser modificado: sem dataset, um dataset ou campo somente leitura, ou um campo calculado. Padrão False.
Field	TField	Public, somente leitura. O campo vinculado, nil enquanto o vínculo não está ativo.

Métodos

Método	Retorna	Descrição
SaveToPNG, SaveToSVG		Como no seletor de hora. Os demais métodos do controle base também se aplicam.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O valor mudou, pelo usuário, em código ou porque o registro se moveu.
OnDropDown, OnCloseUp, OnParseError		Como no seletor de hora.

Como o campo é editado

- O valor acompanha o registro. Um campo nulo mostra um texto vazio.
- Uma alteração do usuário coloca o dataset em modo de edição e marca o valor como modificado. Definir o valor em código também é uma alteração. Quando o dataset não consegue entrar no modo de edição, o valor do campo retorna.
- Enter e a perda de foco gravam o campo, e o dataset grava um valor pendente antes de confirmar. Um campo que recusa o valor gera uma exceção e o foco permanece no controle, como no TDBEdit.
- Escape restaura o valor do campo.
- O seletor de hora mantém a parte de data de um TDateTimeField e altera apenas a hora.

Exemplo: a hora de início de uma 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;
```

Veja também

- Seletor de hora
- Seletor de data DB
- Início rápido data-aware
- TsgcEditorFieldLink

Spin edit DB

Classes: TsgcVCLDBSpinEdit (unit sgcEditors_VCL_DB) e TsgcFMXDBSpinEdit (unit sgcEditors_FMX_DB)
O spin edit DB vincula o valor de um spin edit a um campo numérico. Descende de TsgcVCLSpinEdit e TsgcFMXSpinEdit, de modo que mantém os limites, os passos, as unidades e os botões. Um campo inteiro o muda para esvInteger. As demais propriedades, métodos e eventos publicados são os do spin edit.
Os controles data-aware fazem parte das edições Registered e Trial. Suas units ficam vazias na Community. Eles se vinculam por meio de um TsgcEditorFieldLink, que usa apenas a unit DB, de modo que os controles FireMonkey não precisam de LiveBindings.

Propriedades

Propriedade	Tipo	Descrição
DataSource	TDataSource	A fonte de dados do dataset.
DataField	string	O nome do campo ao qual o valor está vinculado.
AllowEmpty	Boolean	Padrão True, de modo que um campo nulo mostra um texto vazio e apagar o texto grava nulo.
Value	Double	O valor do campo. Não é guardado no formulário.
ReadOnly	Boolean	True mantém o controle somente leitura. Ele também é somente leitura enquanto o campo não pode ser modificado: sem dataset, um dataset ou campo somente leitura, ou um campo calculado. Padrão False.
Field	TField	Public, somente leitura. O campo vinculado, nil enquanto o vínculo não está ativo.

Métodos

Método	Retorna	Descrição
SaveToPNG, SaveToSVG		Como no spin edit. Os demais métodos do controle base também se aplicam.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O valor mudou, pelo usuário, em código ou porque o registro se moveu.
OnParseError, OnValidate		Como no spin edit. OnValidate é executado antes de o valor ir para o campo.

Como o campo é editado

- O valor acompanha o registro. Um campo nulo mostra um texto vazio.
- Uma alteração do usuário coloca o dataset em modo de edição e marca o valor como modificado. Definir o valor em código também é uma alteração. Quando o dataset não consegue entrar no modo de edição, o valor do campo retorna.
- Enter e a perda de foco gravam o campo, e o dataset grava um valor pendente antes de confirmar. Um campo que recusa o valor gera uma exceção e o foco permanece no controle, como no TDBEdit.
- Escape restaura o valor do campo.
- Um campo inteiro muda ValueType para esvInteger.

Exemplo: uma quantidade

```
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;
```

Exemplo: um desconto no 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;
```

Veja também

- Spin edit
- Avaliação DB
- Início rápido data-aware
- TsgcEditorFieldLink

Avaliação DB

Classes: TsgcVCLDBRating (unit sgcEditors_VCL_DB) e TsgcFMXDBRating (unit sgcEditors_FMX_DB)
A avaliação DB vincula o valor de uma avaliação a um campo numérico. Descende de TsgcVCLRating e TsgcFMX-Rating, de modo que mantém Count, Precision, os glifos e as cores. Com Precision erpExact e ReadOnly True, mostra uma média, como 3.7, de um campo calculado ou agregado. As demais propriedades, métodos e eventos publicados são os da avaliação.
Os controles data-aware fazem parte das edições Registered e Trial. Suas units ficam vazias na Community. Eles se vinculam por meio de um TsgcEditorFieldLink, que usa apenas a unit DB, de modo que os controles FireMonkey não precisam de LiveBindings.

Propriedades

Propriedade	Tipo	Descrição
DataSource	TDataSource	A fonte de dados do dataset.
DataField	string	O nome do campo ao qual o valor está vinculado.
Value	Double	O valor do campo. Não é guardado no formulário.
ReadOnly	Boolean	True mantém o controle somente leitura. Ele também é somente leitura enquanto o campo não pode ser modificado: sem dataset, um dataset ou campo somente leitura, ou um campo calculado. Padrão False.
Field	TField	Public, somente leitura. O campo vinculado, nil enquanto o vínculo não está ativo.

Métodos

Método	Retorna	Descrição
SaveToPNG, SaveToSVG		Como na avaliação. Os demais métodos do controle base também se aplicam.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O valor mudou, pelo usuário, em código ou porque o registro se moveu.
OnHover, OnDrawGlyph		Como na avaliação.

Como o campo é editado

- O valor acompanha o registro. Um campo nulo não preenche nenhum glifo.

- Uma alteração do usuário coloca o dataset em modo de edição e marca o valor como modificado. Definir o valor em código também é uma alteração. Quando o dataset não consegue entrar no modo de edição, o valor do campo retorna.
- Enter e a perda de foco gravam o campo, e o dataset grava um valor pendente antes de confirmar. Um campo que recusa o valor gera uma exceção e o foco permanece no controle, como no TDBEdit.
- Escape restaura o valor do campo.
- Com AllowClear, um clique no valor atual grava 0 no campo.

Exemplo: estrelas e uma média

```
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;
```

Veja também

- Avaliação
- Spin edit DB
- Início rápido data-aware
- TsgcEditorFieldLink

Controle deslizante de intervalo DB

Classes: TsgcVCLDBRangeSlider (unit sgcEditors_VCL_DB) e TsgcFMXDBRangeSlider (unit sgcEditors_FMX_DB)
O controle deslizante de intervalo DB vincula Value de um controle deslizante de intervalo a DataField e, em RangeMode, EndValue a EndDataField, de modo que um controle edita um intervalo guardado em dois campos. Descende de TsgcVCLRangeSlider e TsgcFMXRangeSlider. As demais propriedades, métodos e eventos publicados são os do controle deslizante de intervalo.
Os controles data-aware fazem parte das edições Registered e Trial. Suas units ficam vazias na Community. Eles se vinculam por meio de um TsgcEditorFieldLink, que usa apenas a unit DB, de modo que os controles FireMonkey não precisam de LiveBindings.

Propriedades

Propriedade	Tipo	Descrição
DataSource	TDataSource	A fonte de dados do dataset.
DataField	string	O nome do campo ao qual o valor está vinculado.
EndDataField	string	O campo de EndValue em RangeMode.
Value, EndValue	Double	Os valores dos dois campos. Não são guardados no formulário.
EndField	TField	Public, somente leitura. O campo de EndValue, nil enquanto o vínculo não está ativo.
ReadOnly	Boolean	True mantém o controle somente leitura. Ele também é somente leitura enquanto o campo não pode ser modificado: sem dataset, um dataset ou campo somente leitura, ou um campo calculado. Padrão False.
Field	TField	Public, somente leitura. O campo vinculado, nil enquanto o vínculo não está ativo.

Métodos

Método	Retorna	Descrição
SaveToPNG, SaveToSVG		Como no controle deslizante de intervalo. Os demais métodos do controle base também se aplicam.

Eventos

Evento	Tipo	Descrição
OnChange	TNotifyEvent	O valor mudou, pelo usuário, em código ou porque o registro se moveu.
OnChangeComplete, OnFormatLabel		Como no controle deslizante de intervalo.

Como o campo é editado

- O valor acompanha o registro. Um campo nulo coloca o indicador em Min, e um campo final nulo coloca o indicador superior em Max.
- Uma alteração do usuário coloca o dataset em modo de edição e marca o valor como modificado. Definir o valor em código também é uma alteração. Quando o dataset não consegue entrar no modo de edição, o valor do campo retorna.
- Enter e a perda de foco gravam o campo, e o dataset grava um valor pendente antes de confirmar. Um campo que recusa o valor gera uma exceção e o foco permanece no controle, como no TDBEdit.
- Escape restaura o valor do campo.
- Em RangeMode o controle é somente leitura enquanto o campo final não pode ser modificado.

Exemplo: uma faixa de preço em dois 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;
```

Veja também

- Controle deslizante de intervalo
- Avaliação DB
- Início rápido data-aware
- TsgcEditorFieldLink

TsgcEditorModel

Unit: sgcEditors_Types
TsgcEditorModel é a classe base de todo modelo do sgcEditors. É um TPersistent que usa apenas a RTL e o sgcUICore, de modo que funciona com a VCL, com o FireMonkey e sem nenhuma janela, no Windows e no Linux64. Um modelo guarda o valor e as opções, recebe as teclas e o mouse e se desenha em duas camadas: Render desenha a camada de conteúdo e RenderOverlay o hover, o anel de foco e as partes arrastadas. Todo controle VCL e FireMonkey hospeda um modelo, public como Model e como uma propriedade tipada.

Modelo	Unit	Usado por	Propriedade do controle
TsgcCalendarModel	sgcEditors_Calendar	O calendário mensal e a lista suspensa dos seletores de data.	CalendarModel, DropDownCalendar
TsgcDatePickerModel	sgcEditors_Picker	Os seletores de data, hora e data e hora.	PickerModel
TsgcDateTimeDropDownModel	sgcEditors_Picker	A lista suspensa do seletor de data e hora: um calendário e uma lista de horas lado a lado.	
TsgcEditorListModel	sgcEditors_List	A lista de horas e a lista de preenchimento automático do token edit.	DropDownTimeList, TokenModel.AutoCompleteList
TsgcSliderModel	sgcEditors_Slider	O controle deslizante de intervalo.	SliderModel
TsgcColorPickerModel	sgcEditors_Color	O seletor de cor.	ColorModel
TsgcTokenModel	sgcEditors_Token	O token edit.	TokenModel
TsgcRatingModel	sgcEditors_Rating	A avaliação.	RatingModel
TsgcSpinModel	sgcEditors_Spin	O spin edit.	SpinModel

Um modelo publica as mesmas propriedades de valor que seu controle, como Date, Value ou Tokens, além de membros para os hosts, como HitTest e os retângulos de suas partes.

Propriedades

Propriedade	Tipo	Descrição
Theme	TsgcUITheme	utLight (padrão) ou utDark. Veja Temas.
AccentColor	TsgcUIColor	O destaque ARGB. 0 (padrão) usa o destaque do tema.
TextStyle	TsgcUITextStyle	Size, Bold e Color do texto.

Enabled	Boolean	Um modelo desabilitado não recebe entrada e desenha as cores de desabilitado. Padrão True.
ReadOnly	Boolean	O modelo não recebe entrada que altere o valor. Padrão False.
Focused	Boolean	Public. Desenha o anel de foco no overlay. Os hosts o definem quando o controle ganha e perde o foco.
HotPart	Integer	Public. A parte sob o mouse, 0 quando não há nenhuma. Cada modelo numera suas próprias partes.
Painter	TsgcEditors-Painter	Public, somente leitura. As cores do tema e o desenho de moldura, anel de foco, divisa e seta de spin da última renderização.

Métodos

Método	Retorna	Descrição
BeginUpdate, EndUpdate		Agrupa várias alterações. O EndUpdate do par externo envia cada tipo de alteração pendente uma vez.
IsUpdating	Boolean	True entre BeginUpdate e EndUpdate.
KeyDown(var aKey: Word; aShift)	Boolean	Uma tecla, como código de tecla virtual do Windows. As constantes CS_EDITORS_KEY_* guardam os códigos, e o FireMonkey usa os mesmos valores. True quando o modelo a tratou.
KeyPress(var aKey: Char)	Boolean	Um caractere. True quando o modelo o tratou.
MouseDown, MouseUp(aButton, aShift, aX, aY)	Boolean	Um botão do mouse, embLeft, embRight ou embMiddle, em aX e aY em pixels lógicos. True quando o modelo o tratou.
MouseMove(aShift, aX, aY), MouseWheel(aShift, aWheelDelta, aX, aY), MouseLeave	Boolean	O ponteiro e a roda, em pixels lógicos. Um clique da roda vale 120.
Render(aCanvas, aWidth, aHeight, aScale = 1)		Desenha a camada de conteúdo em qualquer TsgcUICanvas. aWidth e aHeight são pixels de dispositivo.
RenderOverlay(aCanvas, aWidth, aHeight, aScale = 1)		Desenha o overlay sobre a camada de conteúdo.
SaveToPNG(aStream or aFileName, aWidth, aHeight, aScale = 1)		Desenha as duas camadas em um PNG. aWidth e aHeight são pixels lógicos e aScale os multiplica.
SaveToSVG(aStream or aFileName, aWidth, aHeight, aScale = 1)		O mesmo em um SVG.
MeasureText(aText, aFont)	Double	A largura de um texto em pixels de dispositivo: OnMeasure quando atribuído, senão a fonte incorporada no sgcUICore.
Assign(aSource)		Copia as opções e o valor de outro modelo da mesma classe.

Eventos

Evento	Tipo	Descrição
OnModelChange	TsgcEditorModelChangeEvent	(Sender; aKind: TsgcEditorChangeKind). eckContent: desenhar a camada de conteúdo novamente. eckOverlay: desenhar apenas o overlay. eckValue: o valor mudou. eckLayout: as partes se moveram, então posicione o edit interno novamente.
OnMeasure	TsgcEditorMeasureEvent	function (const aText: string; const aFont: TsgcUIFontStyle): Double. Mede um texto com a fonte do host, em pixels de dispositivo.

Um controle atribui OnModelChange e OnMeasure do seu próprio modelo. Atribua-os apenas em um modelo que você mesmo cria.

Tamanho e entrada

Um modelo se dispõe e faz o hit test em pixels lógicos, no tamanho da última Render ou no tamanho dado a SetViewSize (Width e Height em TsgcCalendarModel). Chame Render uma vez, ou informe o tamanho, antes de enviar entrada de mouse a um modelo que você mesmo hospeda.

Exemplo: controlando um modelo de controle deslizante sem janela

```
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;
```

Veja também

- Arquitetura
- Saída sem janela
- TsgcEditorLocale

TsgcEditorLocale

Unit: sgcEditors_Date
TsgcEditorLocale guarda os nomes de dias e meses, os formatos de data e hora, os separadores e o primeiro dia da semana que os modelos de data usam para desenhar e interpretar. Um controle mantém uma localidade própria e a preenche a partir do seu FormatSettings, de modo que em um formulário você define FormatSettings, veja Localização. Use TsgcEditorLocale com um modelo que você mesmo cria, como um TsgcCalendarModel sem janela, e com as funções de data da unit.
TsgcCalendarModel e TsgcDatePickerModel têm uma propriedade Locale. O modelo não é dono da localidade, e nil (padrão) usa sgcEditorDefaultLocale, uma localidade compartilhada capturada da localidade do usuário no primeiro uso. Preencha a localidade primeiro e atribua-a depois, porque o modelo desenha novamente quando Locale é atribuído, não quando uma propriedade da localidade muda.

Propriedades

Propriedade	Tipo	Descrição
ShortDayNames[aDay], LongDayNames[aDay]	string	Os nomes de um TsgcEditorWeekDay, de ewdMonday a ewdSunday.
ShortMonthNames[aMonth], LongMonthNames[aMonth]	string	Os nomes de um mês, de 1 a 12.
ShortDateFormat, LongDateFormat	string	Os formatos de data. Um DisplayFormat vazio de um seletor de data usa ShortDateFormat.
ShortTimeFormat, LongTimeFormat	string	Os formatos de hora. Um DisplayFormat vazio de um seletor de hora usa ShortTimeFormat.
TimeAMString, TimePMString	string	Os textos AM e PM de um formato de 12 horas.
DateSeparator, TimeSeparator	Char	Escritos onde um formato tem / e :.
FirstDayOfWeek	TsgcEditorWeekDay	O primeiro dia da semana que efdLocale representa.

Métodos

Método	Retorna	Descrição
Create		Cria a localidade e chama Capture.
Capture		Toma os nomes, os formatos e os separadores da RTL, e o primeiro dia da semana do sistema operacional, segunda-feira quando ele não tem nenhum.
CaptureSettings(aSettings: TFormatSettings)		Toma os nomes, os formatos e os separadores de aSettings, como TFormatSettings.Create('de-DE'). FirstDayOfWeek é mantido.
Assign(aSource)		Copia outra localidade.

Funções

Função	Retorna	Descrição
sgcEditorDefaultLocale	TsgcEditor-Locale	A localidade compartilhada que os modelos usam quando seu Locale é nil. Não a libere.
sgcEditorFormatDateTime(aDateTime, aFor-mat, aLocale = nil)	string	Escreve uma data e hora com um Display-Format e os nomes de uma localidade.
sgcEditorParseDateTime(aText, aFormat, aReference, out aDateTime, aLocale = nil)	Boolean	A interpretação tolerante dos seletores. Os campos que faltam no texto vêm de aRefe-rence.
sgcEditorFirstDayOf(aFirstDay, aLocale = nil)	TsgcEdi-torWeek-Day	Resolve um TsgcEditorFirstDayOfWeek, de modo que efdLocale dá o FirstDayOfWeek da localidade.

Eventos

TsgcEditorLocale não tem eventos.

Exemplo: um calendário alemão em um 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;
```

Veja também

- Localização
- Calendário mensal
- TsgcEditorModel

TsgcEditorFieldLink

Unit: sgcEditors_Data
TsgcEditorFieldLink é o TDataLink dos controles data-aware. Usa apenas a unit DB, de modo que os controles VCL e FireMonkey o compartilham; o FireMonkey não tem TFieldDataLink. Ele traz a semântica do TFieldDataLink, de modo que você também pode usá-lo para vincular um controle seu a um campo. Faz parte das edições Registered e Trial, e a unit fica vazia na Community.

Propriedades

Propriedade	Tipo	Descrição
DataSource	TDataSource	Herdado de TDataLink. A fonte de dados do dataset.
FieldName	string	O nome do campo. É resolvido com FieldByName quando o vínculo se torna ativo e quando os campos do dataset mudam.
Field	TField	Somente leitura. O campo, nil enquanto o vínculo não está ativo ou FieldName está vazio.
ReadOnly	Boolean	Herdado de TDataLink. True impede que o vínculo edite.
CanModify	Boolean	Somente leitura. O vínculo não é ReadOnly, o campo existe e TField.CanModify é True: o campo não é somente leitura nem calculado e o dataset pode modificar.
Editing	Boolean	Somente leitura. True enquanto o dataset está em estado de edição e o campo pode ser modificado.
IsModified	Boolean	Somente leitura. True após Modified, até UpdateRecord gravar o campo, OnDataChange recarregá-lo ou Editing mudar.

Métodos

Método	Retorna	Descrição
Edit	Boolean	Coloca o dataset em modo de edição por meio da fonte de dados, quando o campo pode ser modificado. True quando o vínculo está editando depois disso.
Modified		Marca o valor do controle como alterado.
Reset		Recarrega o controle a partir do campo por meio de OnDataChange e limpa IsModified.
UpdateRecord		Herdado de TDataLink. Dispara OnUpdateData quando IsModified, de modo que o controle grava o campo uma vez. O dataset o chama antes de um post.

Eventos

Evento	Tipo	Descrição
OnDataChange	TNotifyEvent	O registro ou o campo mudou: uma rolagem, um cancelamento, um post, uma atualização ou Reset. Mostre o valor do campo. Não é disparado enquanto Edit coloca o dataset em modo de edição, porque o controle então guarda a alteração pendente do usuário.
OnUpdateData	TNotifyEvent	Grave o valor do controle em Field.
OnActiveChange	TNotifyEvent	O dataset abriu ou fechou.
OnEditingChange	TNotifyEvent	Editing mudou. IsModified é limpo.

Exemplo: vinculando um controle seu

```
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;
```

Veja também

- Início rápido data-aware
- Avaliação DB
- TsgcEditorModel

Entrada de texto e IME

Todo campo de texto do sgcEditors é um controle de edição real: o texto de data ou hora dos seletores, o número do spin edit, o campo hexadecimal do seletor de cor e a entrada após o último chip do token edit. O calendário mensal, o controle deslizante de intervalo e a avaliação não têm texto e são controles desenhados que recebem o foco por conta própria.

Por que um edit nativo

O edit interno é um TEdit filho sem borda na VCL e um TEdit do FMX no FireMonkey, posicionado onde o modelo desenha seu texto. Por ser o edit da plataforma, a composição de IME para entrada em japonês, chinês e coreano, a área de transferência, o desfazer, o menu de contexto e os leitores de tela funcionam sem código especial. A moldura, os botões, as unidades e os chips ao redor são desenhados pelo modelo. Na VCL o edit nunca é uma parada de tabulação própria: o controle repassa o foco a ele, e o anel de foco é desenhado enquanto o edit tem o foco.
No Windows, o canvas raster de cada controle carrega uma vez os arquivos de fonte da plataforma e o edit interno usa a mesma face e o mesmo tamanho em pixels, de modo que o texto digitado e o texto desenhado coincidem. As outras plataformas e a saída sem janela usam Roboto, a fonte incorporada no sgcUICore.

Digitando uma data ou uma hora

- O texto segue DisplayFormat, composto pelos segmentos d, dd, ddd, dddd, m, mm, mmm, mmmm, yy, yyyy, h, hh, n, nn, s, ss, am/pm e ampm mais literais. Um `/` escreve o separador de data da localidade e um `:` o separador de hora. Um DisplayFormat vazio usa a data curta, a hora curta ou as duas, a partir do FormatSettings do controle.
- A interpretação é tolerante. Qualquer um de `/ - . : ,` ou um espaço separa os campos, os nomes de meses e dias da localidade são entendidos, os anos de dois dígitos têm pivô em 50, e um ano de quatro dígitos no início é lido como ano, mês, dia. Um campo que falta no texto toma o ano de hoje e a hora como 0, e um formato sem campos de hora mantém a hora do valor atual.
- Os seletores interpretam o texto a cada alteração e movem junto o calendário aberto. O valor é confirmado com Enter, com a perda de foco e quando a lista suspensa fecha.
- Up e Down avançam o segmento sob o cursor e o selecionam novamente, de modo que você pode girar o mês ou a hora. Os campos dão a volta como o seletor de data e hora do Windows, e depois MinDate, MaxDate, MinTime e MaxTime limitam o resultado.
- Um texto que não pode ser interpretado dispara OnParseError com o texto, o valor e aAccept definido como False. Defina aValue e aAccept para aceitar o texto, ou deixe e o último valor válido retorna. Uma data fora dos limites ou desabilitada no calendário é recusada na confirmação.
- Com AllowEmpty definido como True, um texto vazio limpa o valor e IsEmpty é True. TextHint aparece enquanto o edit está vazio: a partir do Delphi 2009 na VCL, e como o TextPrompt do edit no FireMonkey.

Digitando um número

O spin edit também interpreta com tolerância. Espaços e o separador de milhar da localidade são ignorados, e tanto `.` quanto `,` são aceitos como separador decimal. O texto é formatado com DisplayFormat, uma máscara FormatFloat, ou com Decimals quando DisplayFormat está vazio. Units é um sufixo discreto desenhado após o texto, que nunca faz parte dele. OnParseError funciona como nos seletores, e OnValidate permite alterar ou recusar cada novo valor, com aAccept chegando como True.

Digitando tokens

- Cada caractere de Delimiters, uma vírgula e um ponto e vírgula por padrão, encerra um token, e Enter sempre encerra. Tab e a perda de foco também confirmam o texto digitado. Colar `a, b, c` cria três chips.
- OnValidateToken recebe o texto aparado. Altere-o para normalizar o token, ou limpe aValid para recusá-lo. Com InvalidTokens definido como etiReject o token não é adicionado, e com etiMark ele é adicionado e desenhado como inválido.

- AllowDuplicates, CaseSensitive e MaxTokens limitam o que é adicionado. AddToken, RemoveToken e Clear alteram os tokens em código, e OnTokenAdded e OnTokenRemoved informam cada alteração.

Exemplo

```
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;
```

Veja também

- Teclado e acessibilidade
- Localização
- Seletor de data
- Token edit

Teclado e acessibilidade

Nenhuma função do sgcEditors exige um mouse. Os controles desenhados recebem o foco por conta própria, e os controles com texto repassam o foco ao seu edit interno. Um controle focalizado desenha um anel de foco de 2 pixels na cor de destaque, na sua camada de overlay.

Teclas

Controle	Teclas
Calendário mensal	As setas movem em um dia e em uma semana e ignoram os dias desabilitados. PageUp e PageDown movem em um mês, com Ctrl em um ano. Enter ou Espaço seleciona. Shift com as setas estende um intervalo. Ctrl+Up reduz para a visão de ano e de década, e Enter amplia de volta.
Seletores	F4 ou Alt+Down abre e fecha a lista suspensa. Up e Down avançam o segmento sob o cursor, ou se movem na lista suspensa aberta. Enter confirma, Escape fecha a lista suspensa e restaura o valor.
Spin edit	Up e Down avançam por Increment, PageUp e PageDown por LargeIncrement. A roda do mouse avança por Increment, e um botão mantido pressionado repete mais rápido quanto mais tempo é mantido.
Controle deslizante de intervalo	As setas avançam por Step, PageUp e PageDown por LargeStep, Home e End vão para Min e Max. No modo de intervalo, veja os modelos de foco abaixo.
Avaliação	As setas alteram o valor em um passo de Precision, Home e End vão para o menor valor e para Count, e um dígito define esse valor. Com AllowClear, 0 ou escolher o valor atual novamente o limpa.
Seletor de cor	As setas se movem entre as amostras ou movem o cursor do quadrado e das barras, Enter ou Espaço seleciona, Home e End vão aos extremos. O edit hexadecimal aceita textos #RGB, #RGBA, #RRGGBB e #RRGGBBAA.
Token edit	Enter ou um delimitador adiciona o token digitado. Backspace em uma entrada vazia focaliza o último chip, e um segundo Backspace o remove. Left e Right percorrem os chips, Delete remove o chip focalizado, Escape volta à entrada. Up e Down se movem na lista de preenchimento automático.

Modelos de foco

A maioria dos controles é uma única parada de tabulação. Dois controles têm duas partes, e a VCL e o FireMonkey chegam a elas de formas diferentes, porque o FireMonkey move o foco com Tab antes de o controle ver a tecla.

Controle	VCL	FireMonkey
Controle deslizante de intervalo no modo de intervalo	Tab vai do indicador inferior ao superior, o Tab seguinte sai do controle. Shift+Tab chega ao indicador superior. Ctrl+Left ou Ctrl+Down seleciona o indicador inferior, Ctrl+Right ou Ctrl+Up o superior.	Uma única parada de tabulação. O foco começa no indicador inferior, e um clique mantém o indicador que foi agarrado. Ctrl+Left ou Ctrl+Down seleciona o indicador inferior, Ctrl+Right ou Ctrl+Up o superior.

Seletor de cor	Duas paradas de tabulação: as partes e depois o edit hexadecimal. Enter nas partes seleciona a cor e não chega ao botão padrão do formulário. Um seletor somente leitura é uma única parada de tabulação.	Uma única parada de tabulação com duas paradas internas. Ctrl com as setas alterna entre as partes e o edit hexadecimal.

No FireMonkey o foco é enfileirado pelo framework, e OnEnter e OnExit disparam uma vez por controle, mesmo quando o foco se move entre o controle e seu edit interno.

Acessibilidade

- Os edits internos são edits nativos, de modo que os leitores de tela leem e editam seu texto como de costume.
- Os controles VCL desenhados respondem a WM_GETTEXT com `AccessibleName: value` por meio do proxy de janela do MSAA, e toda mudança de valor chama NotifyWinEvent com EVENT_OBJECT_VALUECHANGE, de modo que um leitor de tela anuncia uma nova posição do controle deslizante ou uma nova avaliação.
- Os controles FireMonkey colocam o mesmo texto em seu Hint, preservando um Hint seu.
- O texto dos dois temas internos tem um contraste de pelo menos 4.5:1, e as cores derivadas de um destaque personalizado são ajustadas para mantê-lo. O QA verifica isso.

Exemplo

```
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;
```

Veja também

- Entrada de texto e IME
- Temas
- Controle deslizante de intervalo
- Seletor de cor

Temas

Todo controle tem um Theme, um AccentColor e um TextStyle. As cores da moldura, do hover, do estado pressionado, da seleção e do estado desabilitado são derivadas deles por TsgcEditorsPainter, de modo que uma propriedade muda o aspecto do controle inteiro.

Propriedade	Efeito
Theme	utLight (padrão) ou utDark, os conjuntos de cores claro e escuro do sgcUICore. utCustom usa o conjunto claro.
AccentColor	A cor da seleção, dos indicadores, do preenchimento e do anel de foco, como um TsgcUIColor ARGB, por exemplo $FF2A78D6. 0 usa o destaque do tema, e uma cor sem alfa é tomada como opaca.
TextStyle	Size (11 por padrão), Bold e Color do texto. Um Color 0 segue a cor de texto do tema.
Enabled	Um controle desabilitado mistura sua superfície, texto, borda e destaque com o fundo.

Cores derivadas

Os preenchimentos de hover e de pressionado são misturas claras do destaque sobre a superfície. O texto sobre um preenchimento de destaque, como o dia selecionado, é uma cor escura ou clara, a que ler melhor, e o preenchimento é escurecido ou clareado até que esse texto alcance um contraste de 4.5:1. A faixa de seleção de um intervalo mantém o mesmo contraste com o texto normal. Assim, qualquer cor de destaque continua legível nos dois temas.

Cores dos controles

- A avaliação tem FillColor e EmptyColor. Um FillColor 0 é âmbar e um EmptyColor 0 segue o tema. Glyph escolhe ergStar, ergHeart, ergCircle ou ergCustom, que desenha por meio de OnDrawGlyph.
- O seletor de cor mostra Swatches, por padrão 24 cores das paletas do sgcUICore e tons de cinza.

A largura de um chip de token depende do tema e da fonte, de modo que uma mudança de Theme ou TextStyle mede todos os chips novamente e dispõe o controle de novo.

Exemplo

```
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;
```

Veja também

- Teclado e acessibilidade
- Saída sem janela

Localização

Cada controle tem uma propriedade pública FormatSettings. Ela começa com as configurações da localidade do usuário, e atribuí-la altera os nomes, os formatos e os separadores que o controle usa. Cada controle mantém sua própria cópia, de modo que dois controles de um mesmo formulário podem usar duas localidades.

Datas e horas

- O calendário e os seletores capturam de FormatSettings os nomes curtos e longos dos dias, os nomes curtos e longos dos meses, os formatos curtos e longos de data e hora, as strings AM e PM e os separadores de data e hora.
- FirstDayOfWeek é efdLocale por padrão, o que toma o primeiro dia da semana do sistema operacional, segunda-feira quando o sistema não tem nenhum. efdMonday, efdSaturday e efdSunday o forçam.
- WeekNumbering é ewnISO8601 por padrão: a semana 1 é a semana que contém a primeira quinta-feira do ano. ewnSimple conta a semana 1 como a semana que contém 1 de janeiro, começando no primeiro dia da semana.
- DisplayFormat vazio significa a data curta da localidade para o seletor de data, a hora curta para o seletor de hora, e as duas, separadas por um espaço, para o seletor de data e hora. A interpretação entende os nomes de meses e dias da localidade. Veja Entrada de texto e IME.

Números

O spin edit e os rótulos e a dica de valor do controle deslizante de intervalo formatam seus números com os separadores decimal e de milhar de FormatSettings. DisplayFormat do spin edit e LabelFormat do controle deslizante são máscaras FormatFloat, e OnFormatLabel do controle deslizante pode escrever qualquer texto, como uma moeda ou uma unidade.

Texto dos controles

O título do calendário, os nomes dos dias e a lista de horas vêm da localidade, e as unidades, as dicas e os nomes acessíveis são propriedades suas.

Exemplo

Um calendário alemão com números de semana ISO, um seletor de data e hora nos formatos curtos alemães, e um spin edit e um controle deslizante com separadores alemães:

```
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;
```

Veja também

- Entrada de texto e IME
- DPI
- TsgcEditorLocale

DPI

O sgcEditors dispõe todo controle em pixels lógicos, pixels a 96 DPI, e o desenha na escala da tela. Nada precisa ser feito no seu código.

VCL

- A escala de um controle é seu PPI dividido por 96. O conteúdo e o overlay são rasterizados nessa escala, e as posições do mouse são divididas por ela antes de chegarem ao modelo.
- Quando o formulário passa para um monitor com outro DPI, ChangeScale e WM_DPICHANGED movem e dimensionam o edit interno e sua fonte junto com o controle.
- A lista suspensa é desenhada no PPI do monitor em que abre, de modo que fica nítida quando o formulário ocupa dois monitores. O Delphi 7 usa Screen.PixelsPerInch.

FireMonkey

O raster de um controle é sua Width multiplicada pela escala da cena, de modo que fica nítido em telas Retina e Android de DPI alto. O modelo mantém seu layout lógico.

Saída sem janela

SaveToPNG e SaveToSVG recebem o tamanho em pixels lógicos e uma escala. Uma escala 2 grava um PNG com o dobro da largura e da altura, o mesmo layout de um controle em uma tela de 192 DPI.

Texto medido

As larguras de texto, como a largura de um chip de token, dependem da escala. O token edit mantém um cache de chips medidos, indexado pelo medidor, pela escala e pelo tema, de modo que uma mudança de DPI mede todos os chips novamente e dispõe o controle de novo.

Exemplo

```
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;
```

Veja também

- Saída sem janela
- Temas

Saída sem janela

Os modelos do sgcEditors renderizam sem janela. Use-os para desenhar um controle em um relatório, em um documento ou em uma página web, para fazer um teste de snapshot, ou para renderizar em um servidor, no Windows ou no Linux64.

Modelos

Crie um modelo, como TsgcCalendarModel, TsgcDatePickerModel, TsgcSliderModel, TsgcColorPickerModel, TsgcTokenModel, TsgcRatingModel ou TsgcSpinModel, defina suas propriedades e chame um destes métodos de TsgcEditorModel. Eles desenham a camada de conteúdo e o overlay.

Método	Grava
SaveToPNG(aStream, aWidth, aHeight, aScale)	Um PNG em um stream.
SaveToPNG(aFileName, aWidth, aHeight, aScale)	Um arquivo PNG.
SaveToSVG(aStream, aWidth, aHeight, aScale)	Um SVG em um stream.
SaveToSVG(aFileName, aWidth, aHeight, aScale)	Um arquivo SVG.

aWidth e aHeight são pixels lógicos, e aScale, 1 por padrão, os multiplica. A saída sem janela usa Roboto, a fonte incorporada no sgcUICore, de modo que o mesmo modelo dá os mesmos pixels em qualquer máquina. Os métodos Render e RenderOverlay desenham em qualquer TsgcUICanvas que você possua.

Controles

Todo controle VCL e FireMonkey também tem SaveToPNG e SaveToSVG com um stream ou um nome de arquivo e uma escala. Eles renderizam o controle no seu tamanho atual, com o texto do seu edit interno desenhado pelo modelo.

Linux64

A partir do Delphi 10.2, as units independentes de framework compilam para Linux64, de modo que um serviço Linux pode renderizar o mesmo calendário, controle deslizante ou avaliação que um aplicativo desktop.

Exemplo

```
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;
```

Veja também

- DPI
- Arquitetura
- TsgcEditorModel

Edições

O sgcEditors é compilado a partir do mesmo código-fonte para todas as edições.

Edição	Inclui
Registered	Tudo: os nove controles para VCL e FireMonkey, os cinco controles data-aware de cada framework, o quadrado HSV e a barra alfa do seletor de cor, a lista de preenchimento automático do token edit e os dois indicadores do controle deslizante de intervalo.
Trial	Os mesmos recursos da Registered. Um build compilado com SGC_TRIAL mostra uma mensagem de que é uma versão de avaliação quando um componente é criado depois que passou a janela de lançamento indicada no build. Nada mais muda.
Community	Todos os nove controles para VCL e FireMonkey sem os recursos avançados listados abaixo, e sem aviso de avaliação.

Os recursos avançados

Os recursos avançados são compilados sob o define SGC_EDITORS_ADVANCED, que o sgcVer.inc desativa na edição Community. Isto é exatamente o que ele altera:

Recurso	Na Community
Seletor de cor	ecpHSV e ecpAlpha em Parts são ignorados: o quadrado, a barra de matiz e a barra alfa não são desenhados, o alfa é sempre $FF e o texto hexadecimal é #RRGGBB. As amostras e o edit hexadecimal permanecem.
Token edit	AutoCompleteItems, AutoCompleteMatch, AutoCompleteMinChars e AutoCompleteMaxItems permanecem, de modo que um formulário salvo por um build Registered carrega, mas a lista nunca abre.
Controle deslizante de intervalo	RangeMode permanece, mas um indicador é desenhado e editado. EndValue é mantido e não é mostrado.
Controles data-aware	As units sgcEditors_Data, sgcEditors_VCL_DB e sgcEditors_FMX_DB compilam como units vazias, de modo que as dez classes data-aware e seu registro não existem. A paleta mostra 18 componentes em vez de 28.

A Community mantém o calendário completo, os três seletores com suas listas suspensas, o spin edit com unidades, a avaliação de meia estrela, a validação de tokens, os temas, o teclado, o DPI e a saída PNG e SVG sem janela.

Código para todas as edições

A edição é uma escolha em tempo de compilação no sgcVer.inc, e SGC_COMMUNITY também desativa SGC_TRIAL. Uma unit que precisa compilar na Community e usa um membro que só existe na edição avançada inclui sgcVer.inc e coloca esse código dentro de {$IFDEF SGC_EDITORS_ADVANCED}. Um formulário com um controle data-aware não carrega em um build Community, porque a classe não existe ali. sgcEditors_Const tem a versão e a edição 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;
```

Veja também

- Instalação
- Início rápido data-aware

Visão geral do .NET

sgcEditors .NET é uma versão gerenciada em C# da biblioteca sgcEditors do Delphi, não um wrapper em torno dela. As classes C# ficam no namespace **esegece.sgcEditors** e mantêm os nomes do Delphi, de modo que **TsgcCalendarModel**, **TsgcSpinModel**, **TsgcEditorLocale** e as enumerações são lidos da mesma forma nas duas linguagens, e os nove controles seguem o padrão de nomes do Delphi em todos os hosts: **TsgcVCLDatePicker** é **TsgcWinFormsDatePicker**, **TsgcWpfDatePicker** e **TsgcAvaloniaDatePicker**. Os controles são desenhados com a versão .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe, em torno de uma caixa de texto hospedada do framework para o texto digitado.

Pacotes

A versão é dividida em cinco pacotes NuGet, de modo que um projeto só traz o host que usa.

PackageId	Contém	Frameworks de destino
esegece.sgcEditors	Os modelos dos nove controles (**TsgcCalendarModel**, **TsgcDatePickerModel**, **TsgcSliderModel**, **TsgcRatingModel**, **TsgcSpinModel**, **TsgcColorPickerModel**, **TsgcTokenModel**), os modelos de lista e de lista suspensa, a matemática de datas, a localidade e a saída SVG sem janela. Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcEditors.Rendering	A saída PNG sem janela, os métodos de extensão **SaveToPNG** e **RenderToRaster** de **sgcEditors_Raster**, desenhados no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, Linux e macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcEditors.WinForms	Os nove controles **TsgcWinForms** sobre **TsgcWinFormsCustomEditor**.	net472, net8.0-windows, net9.0-windows
esegece.sgcEditors.Wpf	Os nove controles **TsgcWpf** sobre **TsgcWpfCustomEditor**.	net472, net8.0-windows, net9.0-windows
esegece.sgcEditors.Avalonia	Os nove controles **TsgcAvalonia** sobre **TsgcAvaloniaCustomEditor**, no Windows, Linux e macOS com o Avalonia 11.	net8.0, net9.0

Os três pacotes de host dependem de esegece.sgcEditors.Rendering, que depende de esegece.sgcEditors, portanto uma linha `PackageReference` traz tudo o que um host precisa. Veja Instalação pelo NuGet.

Edições

Edição	Ids dos pacotes	Inclui
Registered	esegece.sgcEditors, .Rendering, .WinForms, .Wpf e .Avalonia	Tudo: os nove controles dos três hosts, os temas, o suporte a teclado e acessibilidade, a saída PNG e SVG sem janela e os três recursos avançados: o quadrado HSV e a barra alfa do seletor de cor, a lista de autocompletar do token edit e o segundo seletor do controle deslizante de intervalo.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcEditors.WinForms.Community.	Tudo, exceto os três recursos avançados. As propriedades permanecem, de modo que o mesmo código compila nas duas edições: **Parts** ignora **ecpHSV** e **ecpAlpha** (o alfa é sempre $FF), **AutoCompleteItems** nunca abre uma lista, e **RangeMode** desenha e edita um único indicador enquanto **EndValue** é mantido.

É a mesma divisão das edições do Delphi, veja Edições. Os controles data-aware do Delphi não têm equivalente no .NET em nenhuma edição: as propriedades de valor são vinculadas com a vinculação de dados de cada framework, veja Vinculação de dados no .NET.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	VCL (controles TsgcVCL)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	VCL e FMX	Windows, .NET Framework 4.7.2 e .NET 8 e 9. Os controles podem ser declarados em XAML.
Avalonia	FMX (controles TsgcFMX)	Windows, Linux e macOS, .NET 8 e 9.
Sem janela	os modelos e seus SaveToPNG e SaveToSVG	Qualquer plataforma dos frameworks de destino: SVG com esegece.sgcEditors, PNG com esegece.sgcEditors.Rendering.

Todo host tem a mesma API: as propriedades, os métodos e os eventos dos controles do Delphi com os mesmos nomes, mais **Theme**, **AccentColor**, **TextStyle**, **ReadOnly**, **FormatSettings**, **DropDown**, **CloseUp**, **SaveToPNG** e **SaveToSVG** na classe base. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Vinculação de dados no .NET
- .NET Headless Rendering
- Demos do .NET
- Plataformas

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, os pacotes do núcleo e da renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcEditors.WinForms
dotnet add package esegece.sgcEditors.Wpf
dotnet add package esegece.sgcEditors.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcEditors.WinForms.Community
dotnet add package esegece.sgcEditors.Wpf.Community
dotnet add package esegece.sgcEditors.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcEditors.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie os pacotes Registered ou os pacotes Community em um projeto, nunca os dois: eles contêm os mesmos tipos no mesmo namespace.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:sgc="clr-namespace:esegece.sgcEditors;assembly=esegece.sgcEditors.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicativo desktop Avalonia. A caixa de texto interna adota a aparência do tema Fluent. Mapeie o namespace no XAML com `xmlns:sgc="using:esegece.sgcEditors"`.
- Sem janela: esegece.sgcEditors sozinho basta para SVG. Adicione esegece.sgcEditors.Rendering para PNG, ele traz o SkiaSharp e seus recursos nativos para Windows e Linux.

Em C#, todo host precisa de `using esegece.sgcEditors;`, e de `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUITextStyle** e os demais tipos dos temas.

Veja também

- Visão geral do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Demos do .NET

Início rápido WinForms

Este início rápido monta um pequeno formulário de reserva em código com quatro dos nove controles WinForms: um seletor de data com calendário suspenso, um spin edit com unidades, um controle deslizante de intervalo e um token edit.

Passos

1. Crie um projeto Windows Forms para `net8.0-windows` e adicione o pacote **esegece.sgcEditors.WinForms** (**esegece.sgcEditors.WinForms.Community** na edição Community).
2. Adicione `using esegece.sgcEditors;` e `using esegece.sgcUICore;`.
3. Crie os controles, defina suas propriedades e adicione-os ao formulário. São controles WinForms comuns: Location, Size, Dock, Anchor, TabIndex e Enabled funcionam como de costume.
4. Atribua os eventos. **OnChange** é disparado por todos os controles quando seu valor muda, os outros eventos são os do controle do Delphi.

Exemplo

Notas

- Os valores de data e hora são datas OLE Automation em um `double`, o TDateTime do Delphi. Converta com `DateTime.ToOADate` e `DateTime.FromOADate`.
- O controle deslizante de intervalo espera entre **BeginInit** e **EndInit** para limitar **Value** e **EndValue** a **Min** e **Max**, portanto a ordem das atribuições não importa. O designer do Windows Forms chama os dois, por meio de `ISupportInitialize`.
- **Theme** é por controle. Todo controle descende de **TsgcWinFormsCustomEditor**, portanto um laço sobre os controles define o tema de um formulário inteiro.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Vinculação de dados no .NET
- Seletor de data

Início rápido WPF

Os controles WPF podem ser declarados em XAML como qualquer outro controle. Este início rápido declara um seletor de data e hora, uma avaliação em meia estrela e um seletor de cor, e define o restante no code behind.

Passos

1. Crie um projeto WPF para `net8.0-windows` e adicione o pacote **esegece.sgcEditors.Wpf** (**esegece.sgcEditors.Wpf.Community** na edição Community).
2. Mapeie o namespace na janela: `xmlns:sgc="clr-namespace:esegece.sgcEditors;assembly=esegece.sgcEditors.Wpf"`.
3. Declare os controles. As enumerações recebem os nomes dos seus valores, como `Precision="erpHalf"`. Um controle sem Width e Height assume um tamanho padrão (160 x 28 para a maioria dos seletores, 200 x 28 para o seletor de data e hora, 280 x 240 para o calendário mensal).
4. Atribua os eventos no code behind.

Exemplo

A janela:

```
<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>
```

O code behind:

Notas

- As propriedades de valor são propriedades de dependência que vinculam nos dois sentidos por padrão: **Date**, **Time** e **DateTime** dos seletores, **Value** e **EndValue** do controle deslizante, **Value** da avaliação e do spin edit, **SelectedColor** do seletor de cor. Veja Vinculação de dados no .NET.
- O seletor de cor tem **SelectedWpfColor**, o `System.Windows.Media.Color` opaco de **SelectedColor**.
- As listas suspensas são popups do WPF posicionados abaixo do controle, ou acima dele perto da parte inferior da tela, e desenhados no DPI do monitor do controle.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Vinculação de dados no .NET
- Avaliação

Início rápido Avalonia

Os controles Avalonia são executados no Windows, Linux e macOS. Este início rápido é um aplicativo Avalonia completo em código: um calendário mensal que seleciona um intervalo de até 14 dias e um seletor de hora com uma lista suspensa a cada 30 minutos.

Passos

1. Crie um projeto de console `net8.0`, defina `<OutputType>WinExe</OutputType>` e adicione **Avalonia.Desktop** e **Avalonia.Themes.Fluent** 11.3 e o pacote **esegece.sgcEditors.Avalonia** (**esegece.sgcEditors.Avalonia.Community** na edição Community).
2. Adicione o tema Fluent ao aplicativo. A caixa de texto interna dos seletores adota a aparência dele.
3. Crie os controles e adicione-os a um painel. No XAML, mapeie o namespace com `xmlns:sgc="using:esegece.sgcEditors"`.

Exemplo

Notas

- Uma hora é a fração de um dia de uma data OLE Automation, portanto `TimeSpan.TotalDays` é um valor de hora.
- As propriedades de valor são propriedades estilizadas que vinculam nos dois sentidos por padrão. Veja Vinculação de dados no .NET.
- A seleção de intervalo faz parte de todas as edições. O segundo indicador do controle deslizante de intervalo é o recurso avançado, veja Visão geral do .NET.
- **AutomationProperties.Name** dá um nome a um controle para os leitores de tela, seu valor é exposto pelo padrão de valor do seu automation peer.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Vinculação de dados no .NET
- Calendário mensal

Classes do .NET

Cada classe do Delphi mantém seu nome no .NET. Os controles VCL e FireMonkey se tornam os controles WinForms, WPF e Avalonia com o prefixo do host, e as classes independentes de framework são iguais em todos os hosts.

Classes dos hosts

Delphi	WinForms	WPF	Avalonia	Model (propriedade do host)
TsgcVCLMonthCalendar, TsgcFMXMonthCalendar	TsgcWinForms-MonthCalendar	TsgcWpfMonth-Calendar	TsgcAvalonia-MonthCalendar	TsgcCalendarMo-del (CalendarMo-del)
TsgcVCLDatePicker, TsgcFMXDatePicker	TsgcWinForms-DatePicker	TsgcWpfDate-Picker	TsgcAvalonia-DatePicker	TsgcDatePicker-Model (PickerMo-del)
TsgcVCLTimePicker, TsgcFMXTimePicker	TsgcWinFormsTi-mePicker	TsgcWpfTime-Picker	TsgcAvaloniaTi-mePicker	TsgcDatePicker-Model (PickerMo-del)
TsgcVCLDateTimePicker, TsgcFMXDateTimePicker	TsgcWinForms-DateTimePicker	TsgcWpfDateTi-mePicker	TsgcAvalonia-DateTimePicker	TsgcDatePicker-Model (PickerMo-del)
TsgcVCLRangeSlider, TsgcFMXRangeSlider	TsgcWinForms-RangeSlider	TsgcWpfRan-geSlider	TsgcAvalonia-RangeSlider	TsgcSliderModel (SliderModel)
TsgcVCLRating, TsgcFMXRating	TsgcWinForms-Rating	TsgcWpfRating	TsgcAvalonia-Rating	TsgcRatingModel (RatingModel)
TsgcVCLSpinEdit, TsgcFMXSpinEdit	TsgcWinFormsS-pinEdit	TsgcWpfSpinE-dit	TsgcAvaloniaS-pinEdit	TsgcSpinModel (SpinModel)
TsgcVCLColorPicker, TsgcFMXColorPicker	TsgcWinForms-ColorPicker	TsgcWpfColor-Picker	TsgcAvalonia-ColorPicker	TsgcColorPicker-Model (ColorMo-del)
TsgcVCLTokenEdit, TsgcFMXTokenEdit	TsgcWinForms-TokenEdit	TsgcWpfToke-nEdit	TsgcAvalonia-TokenEdit	TsgcTokenModel (TokenModel)
TsgcVCLCustomEditor, TsgcFMXCustomEditor	TsgcWinForms-CustomEditor	TsgcWpfCusto-mEditor	TsgcAvalonia-CustomEditor	TsgcEditorModel (Model)
TsgcVCLCustomDatePic-ker, TsgcFMXCustomDa-tePicker	TsgcWinForms-CustomDatePic-ker	TsgcWpfCus-tomDatePicker	TsgcAvalonia-CustomDate-Picker	TsgcDatePicker-Model (PickerMo-del)
o TEdit interno	TsgcWinFormsE-ditorInnerEdit (a TextBox)	TsgcWpfEdito-rInnerEdit (a TextBox)	TsgcAvaloniaE-ditorInnerEdit (a TextBox)	
a lista suspensa	TsgcWinFormsE-ditorPopup (a To-	TsgcWpfEditor-Popup (a Po-pup)	TsgcAvaloniaE-ditorPopup (a Popup)	TsgcDateTime-DropDownModel,

	olStripDrop-Down)			TsgcEditorListMo-del

As classes base derivam de **TsgcUIWinFormsControl**, **TsgcUIWpfControl** e **TsgcUIAvaloniaControl** do sgcUI-Core. As classes de acessibilidade são **TsgcWinFormsEditorAccessibleObject**, **TsgcWpfEditorAutomationPe-er** e **TsgcAvaloniaEditorAutomationPeer**; elas informam o nome e o valor do controle.

Membros de todos os controles

Membro	Descrição
Model	O modelo do controle, do tipo da tabela acima. A propriedade tipada (CalendarMo-del, SpinModel...) retorna o mesmo objeto.
InnerEdit	A caixa de texto hospedada, null nos controles sem texto digitado (calendário men-sal, controle deslizante de intervalo, avaliação).
Popup	A lista suspensa dos seletores e da lista de autocompletar dos tokens.
Theme, AccentColor, TextStyle	A aparência: **TsgcUITheme** (utLight, utDark), a cor de destaque como `uint` ($AARRGGBB, 0 é a cor do tema) e o tamanho e o peso da fonte.
ReadOnly	O valor pode ser lido e copiado, mas não alterado.
FormatSettings	Uma cópia do **TsgcEditorFormatSettings** com o qual o controle formata e interpre-ta, a cultura atual por padrão. Altere a cópia e atribua-a de volta.
DroppedDown, Drop-Down(), CloseUp(bool)	O estado da lista suspensa, abri-la e fechá-la com ou sem aceitar.
OnChange	Disparado quando o valor muda, Sender é o controle.
SaveToPNG, Save-ToSVG	Desenham o controle como ele aparece agora em um stream ou em um arquivo, em uma escala.

Tipos independentes de framework

Tipo	Descrição
TsgcEditorModel	O modelo base: tema, foco, habilitado, hover, teclado, mouse, **Render**, **RenderO-verlay** e **SaveToSVG**. Veja TsgcEditorModel.
TsgcEditorLocale	Os nomes dos dias e dos meses e o primeiro dia da semana, obtidos das configura-ções de formato. Veja TsgcEditorLocale.
TsgcEditorFormat-Settings	O subconjunto do TFormatSettings do Delphi que os editores leem. `Create()` usa a cultura atual, `Create(CultureInfo)` outra.
TsgcEditorStrings	O TStrings do Delphi: **DisabledDates**, **Swatches**, **Tokens**, **AutoCompleteItems**.
TsgcEditorListModel	A lista da lista suspensa de horas e do autocompletar dos tokens.
TsgcDateTimeDrop-DownModel	A lista suspensa do seletor de data e hora: um calendário ao lado de uma lista de horas.

sgcEditors_Types, sgcEditors_Date	Classes estáticas com as funções livres e as constantes do Delphi: códigos de tecla (CS_EDITORS_KEY_*), formatação de números, conversão hexadecimal e HSV, a matemática de datas.
sgcEditors_Raster	Em esegece.sgcEditors.Rendering: os métodos de extensão **SaveToPNG** e **RenderToRaster** de todos os modelos.

Enumerações

As enumerações mantêm os nomes e os prefixos de valor do Delphi: **TsgcEditorSelectionMode** (esm), **TsgcEditorFirstDayOfWeek** (efd), **TsgcEditorWeekNumbering** (ewn), **TsgcEditorCalendarView** (ecv), **TsgcEditorOrientation** (eor), **TsgcEditorTickPlacement** (etp), **TsgcEditorInvalidTokens** (eti), **TsgcEditorMatchMode** (emm), **TsgcEditorRatingPrecision** (erp), **TsgcEditorRatingGlyph** (erg), **TsgcEditorSpinValueType** (esv) e **TsgcEditorSpinButtons** (esb). Os conjuntos do Delphi são enumerações `[Flags]` combinadas com `|`: **TsgcEditorWeekDays**, **TsgcEditorColorParts**, **TsgcEditorChangeKinds** e **TsgcEditorShiftState**.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- Controles
- Arquitetura

Diferenças do .NET em relação ao Delphi

A versão .NET mantém os nomes, os padrões, o comportamento do teclado e as regras de renderização do Delphi. A saída SVG dos modelos é idêntica byte a byte à do Delphi, e a saída PNG coincide com ela dentro da tolerância do rasterizador SkiaSharp. As diferenças abaixo vêm dos frameworks dos hosts e da biblioteca de classes do .NET.

Sem controles data-aware

Não há controles DB nem **TsgcEditorFieldLink** no .NET, em nenhuma edição. As propriedades de valor são vinculadas com a vinculação de dados de cada framework: `Binding` do WinForms por meio de `INotifyPropertyChanged`, propriedades de dependência do WPF e propriedades estilizadas do Avalonia, ambas nos dois sentidos por padrão. Veja Vinculação de dados no .NET.

Sem streaming de componentes

Não há DFM no .NET, portanto não há editor de lista de strings em tempo de projeto para **DisabledDates** e **Swatches**: preencha-as em código. O controle deslizante de intervalo adia o limite de **Value** e **EndValue** para entre **BeginInit** e **EndInit**, o gancho que o designer do Windows Forms e os carregadores de XAML chamam, onde o Delphi espera por Loaded.

Valores, eventos e listas

- TDateTime é um `double`, uma data OLE Automation: `DateTime.ToOADate` e `DateTime.FromOADate` a convertem. Uma hora é a fração de um dia.
- TsgcUIColor é um `uint`, $AARRGGBB, em que 0 significa a cor do tema. **SelectedVCLColor** é **SelectedWinFormsColor** (`System.Drawing.Color`), **SelectedWpfColor** (`System.Windows.Media.Color`) e **SelectedAvaloniaColor** (`Avalonia.Media.Color`), todas opacas.
- Os eventos são propriedades delegate com os mesmos nomes do Delphi (**OnChange**, **OnSelect**, **OnTokenAdded**...). Atribua um único manipulador com `=`. Um parâmetro `var` do Delphi é um parâmetro `ref`, e um lambda com um parâmetro `ref` declara o tipo de todos os parâmetros. **TNotifyEvent** é **TsgcUINotifyEvent**, e Sender é o controle.
- Uma propriedade TStrings é um **TsgcEditorStrings**, com **Add**, **Insert**, **Delete**, **Clear**, **IndexOf** (que ignora maiúsculas e minúsculas), **Assign**, **BeginUpdate** e **EndUpdate**. Os objetos da lista de tokens são acessados com **GetObject**.
- Uma função do Delphi sem parâmetros continua sendo um método, como `IsRange()`, `CellCount()` ou `HexText()`. Um conjunto do Delphi é uma enumeração `[Flags]` combinada com `|`.
- TFormatSettings é **TsgcEditorFormatSettings**. A propriedade **FormatSettings** retorna uma cópia: altere-a e atribua-a de volta, como no record do Delphi.

Caixa de texto, teclado e foco em cada host

- O texto digitado é um `TextBox` hospedado do framework, portanto o cursor, a seleção, a área de transferência, o desfazer e o IME são os do framework. **TextHint** é o texto de dica (cue banner) no WinForms, uma dica desenhada pelo edit interno no WPF e o `Watermark` no Avalonia.
- Tab, Enter e Escape se comportam como no Delphi nos três hosts. No Avalonia, uma tecla tratada pelo controle não chega a um manipulador KeyDown do próprio controle.
- O modelo trabalha em pixels lógicos. WPF e Avalonia fazem o layout em DIPs, portanto o edit interno e a lista suspensa mantêm o mesmo tamanho em qualquer DPI; o WinForms os escala pelo DPI do monitor.
- Acessibilidade: o WinForms informa o nome e o valor por meio do `AccessibleObject`, o WPF por meio de um `AutomationPeer` com o padrão de valor, o Avalonia por meio de `AutomationProperties` e do seu automation peer. Não há texto de janela como na VCL.
- No WPF, **Tokens** do token edit é uma propriedade CLR, não uma propriedade de dependência; **OnChange** informa suas alterações. No Avalonia é uma propriedade direta notificada a cada alteração.

Veja também

- Visão geral do .NET
- Classes do .NET
- Vinculação de dados no .NET
- Teclado e acessibilidade

Vinculação de dados no .NET

Os controles data-aware do Delphi ligam um controle a um campo de um TDataSet por meio de **TsgcEditorFieldLink**. No .NET, as propriedades de valor dos controles comuns são vinculadas com a vinculação de dados de cada framework, nas duas edições, portanto não existem controles DB separados.

Controle	Propriedades de valor
MonthCalendar	Date, RangeStart, RangeEnd
DatePicker, TimePicker, DateTimePicker	Date, Time, DateTime (as três são a mesma data OLE Automation vista como data, hora ou ambas), IsEmpty (somente leitura)
RangeSlider	Value, EndValue
Rating	Value
SpinEdit	Value, IsEmpty (somente leitura)
ColorPicker	SelectedColor (`uint`), mais a cor somente leitura do tipo do host
TokenEdit	Tokens

WinForms

Os controles WinForms implementam `INotifyPropertyChanged` e o disparam para as propriedades de valor, portanto um `Binding` com `DataSourceUpdateMode.OnPropertyChanged` grava de volta na hora, enquanto o usuário digita, incrementa ou arrasta.

WPF

As propriedades de valor são propriedades de dependência registradas para vincular nos dois sentidos por padrão: **DateProperty**, **TimeProperty**, **DateTimeProperty**, **RangeStartProperty**, **RangeEndProperty**, **ValueProperty**, **EndValueProperty**, **SelectedColorProperty** e a **IsEmptyProperty**, somente leitura. O controle pode limitar um valor vinculado aos seus limites e então grava o valor limitado de volta. **Tokens** é uma propriedade CLR e não é vinculada.

Avalonia

As propriedades de valor são propriedades estilizadas que vinculam nos dois sentidos por padrão, com os mesmos nomes do WPF. **IsEmptyProperty**, **SelectedAvaloniaColorProperty** e **TokensProperty** são propriedades diretas: vincule **Tokens** nos dois sentidos a um **TsgcEditorStrings**, o setter copia as strings.
A demo BindingWpf vincula um seletor de data, um seletor de hora, um spin edit, uma avaliação e um controle deslizante de intervalo a uma lista de reservas, veja Demos do .NET.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- Demos do .NET
- Início rápido data-aware

Renderização sem janela no .NET

Todo controle desenha por meio do seu modelo, e os modelos não precisam de janela. Um serviço, um teste ou um relatório pode criar um modelo, definir suas propriedades e salvá-lo como SVG ou PNG em qualquer plataforma, inclusive Linux.

- **SaveToSVG** é um método de **TsgcEditorModel** em esegece.sgcEditors, sem dependência nativa.
- **SaveToPNG** é um método de extensão de **sgcEditors_Raster** em esegece.sgcEditors.Rendering, desenhado com SkiaSharp. **RenderToRaster** retorna o próprio canvas raster; quem chama o libera.

Os dois recebem o stream ou o nome do arquivo, a largura e a altura em pixels lógicos e uma escala opcional: uma escala de 2 grava uma imagem de 560 x 480 para um calendário de 280 x 240. Os dois desenham a camada de conteúdo e depois a camada de sobreposição (hover, anel de foco, um indicador arrastado), de modo que a saída é o que o controle mostra.

Exemplo

Notas

- Os hosts também têm **SaveToPNG** e **SaveToSVG**, com o tamanho do controle.
- Os modelos formatam números e datas com a cultura atual. Para obter a mesma saída em todas as máquinas, atribua `TsgcEditorFormatSettings.Create(CultureInfo.InvariantCulture)` a **FormatSettings** dos modelos spin e slider, ou do controle do host.
- As listas suspensas também são modelos (**TsgcDateTimeDropDownModel**, **TsgcEditorListModel**), desenhados com os mesmos métodos.

Veja também

- Visão geral do .NET
- Classes do .NET
- Saída sem janela

Demos do .NET

As demos estão em `.net\demos\Editors`, uma pasta por demo com seu projeto, seu `Program.cs` e um README.md. Elas montam a interface em código, sem XAML. Toda demo tem a opção `--test`, que executa uma autoverificação fora da tela, escreve linhas PASS e FAIL e termina com o número de verificações que falharam, como a opção `/test` das demos do Delphi.

Demo	Host	Mostra
Gallery	WinForms	Os nove controles em uma janela: um calendário mensal com números de semana, um seletor de data, um seletor de hora com uma lista de 15 minutos, um seletor de data e hora, um spin edit com unidades, uma avaliação, um controle deslizante de intervalo com marcas, um token edit com lista de autocompletar e um seletor de cor com amostras, o quadrado HSV, o alfa e um campo hexadecimal. Um seletor de tema, um seletor de somente leitura, exportação para PNG e um log de eventos.
GalleryWpf	WPF	A mesma janela com os controles **TsgcWpf**.
GalleryAvalonia	Avalonia	A mesma janela com os controles **TsgcAvalonia**, para Windows, Linux e macOS.
BookingForm	WinForms	Um formulário de reserva de hotel com sete controles: um calendário em modo de intervalo para a estadia (no máximo 28 noites), seletores de hora de check-in e check-out com limites e uma lista a cada 30 minutos, um spin edit para os hóspedes, um token edit para até seis pedidos com lista de autocompletar, uma avaliação para a categoria do hotel e um seletor de cor com amostras e hexadecimal para a cor da etiqueta.
BindingWpf	WPF	MVVM: uma lista de reservas em um view model, e um seletor de data, um seletor de hora, um spin edit, uma avaliação e um controle deslizante de intervalo vinculados à reserva selecionada, o equivalente .NET da demo data-aware do Delphi.

As demos Gallery mostram o quadrado HSV, a barra alfa, o segundo seletor do controle deslizante de intervalo e a lista de autocompletar somente na edição Registered, e o BookingForm mostra a lista de autocompletar somente nela. Na edição Community o mesmo código é executado sem eles.

Compilando as demos

Dentro da árvore do produto, as demos são compiladas contra os projetos da biblioteca. O zip das demos usa por padrão os pacotes NuGet da edição Community. Toda demo compila e executa nas duas edições. Compile uma demo com os pacotes Registered definindo **SgcEditorsEdition**:

```
dotnet build Gallery\Gallery.csproj -p:SgcEditorsEdition=Registered
dotnet run --project BookingForm\BookingForm.csproj -- --test
```

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Início rápido WinForms
- Vinculação de dados no .NET

Solução de problemas

Problemas comuns do sgcEditors e suas causas.

Problema	Causa e solução
Uma data digitada volta ao valor anterior.	O texto não pôde ser interpretado, ou a data está fora de MinDate e MaxDate, ou está desabilitada por DisabledDates, DisabledWeekDays ou OnIsDateDisabled. Trate OnParseError para aceitar suas próprias palavras, e chame IsDateEnabled antes de definir uma data em código.
O seletor de hora não tem botão de lista suspensa.	ListInterval é 0. Defina-o como os minutos entre os itens da lista, 30 por padrão. Os itens ficam entre MinTime e MaxTime.
Um valor definido em código é alterado.	O valor é limitado pelos limites. Defina MinValue, MaxValue, ValueType e Decimals antes de Value. MinValue igual a MaxValue significa sem limite. O Max de um controle deslizante com Step desce para a grade de passos, como em um input range do HTML.
As escolhas em uma lista suspensa do FireMonkey se perdem.	Um clique fora da lista suspensa a fecha e desfaz as escolhas feitas enquanto estava aberta. Feche-a com Enter ou com uma escolha.
Tab não vai para o segundo indicador no FireMonkey.	O FireMonkey move o foco com Tab. Use Ctrl+Right ou Ctrl+Up para o indicador superior e Ctrl+Left ou Ctrl+Down para o indicador inferior.
O seletor de cor não tem quadrado nem barra alfa.	Parts não inclui ecpHSV nem ecpAlpha, ou o build é Community, que os ignora.
A lista de preenchimento automático nunca abre.	AutoCompleteItems está vazio, foram digitados menos de AutoCompleteMinChars caracteres, ou o build é Community.
O controle deslizante de intervalo mostra um indicador.	RangeMode é False, ou o build é Community.
Class TsgcVCLDBDatePicker not found when a form loads.	O aplicativo é compilado com a edição Community, que não tem controles data-aware.
Um controle data-aware não pode ser editado.	ReadOnly é True, falta DataSource ou seu dataset, o dataset ou o campo é somente leitura, o campo é calculado ou, para um controle deslizante no modo de intervalo, o campo final não pode ser modificado.
Definir Value de um controle data-aware coloca o dataset em modo de edição.	Um valor definido em código conta como uma alteração do usuário. Para alterar o registro sem editá-lo, grave o campo.
O PNG sem janela usa outra fonte que não a da tela.	A tela usa a fonte da plataforma, Segoe UI no Windows, e a saída sem janela usa o Roboto incorporado, para que a saída seja a mesma em todo lugar.
A entrada IME não funciona no calendá-	Esses controles não têm texto. O IME funciona nos edits internos dos seletores, do spin edit, do seletor de cor e do token edit.

rio, no controle deslizante ou na avaliação.	
Um formulário com centenas de editores abre lentamente.	A maior parte do tempo vai para a criação das janelas dos controles e de seus edits internos: cerca de 340 ms para 270 controles. Crie os controles das páginas ocultas quando a página for mostrada pela primeira vez.

Exemplo

```
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;
```

Veja também

- Entrada de texto e IME
- Início rápido data-aware
- Edições

Introdução

sgcGauges adiciona medidores de instrumento a aplicativos Delphi e C++Builder, para VCL, para FireMonkey e sem nenhum framework de interface de usuário. Existem dez tipos de medidor: radial, linear, digital, LED, barra de LEDs, anel de progresso, VU meter, relógio, bússola e anunciador. Cada tipo existe como um controle VCL com o prefixo TsgcVCL, como um controle FireMonkey com o prefixo TsgcFMX e como uma classe de modelo independente de framework, como TsgcRadialGauge, que gera PNG e SVG sem janela.
Um medidor tem uma ou mais escalas, cada uma com suas faixas de cor, um ou mais ponteiros (agulha, marcador, barra ou faixa) e limiares que disparam um evento quando um valor os cruza. Um novo valor move o ponteiro com uma animação e uma curva de suavização. Os medidores têm temas claro e escuro, se adaptam ao seu tamanho e ao DPI, e o Value de um medidor pode ser escrito a partir de qualquer thread.
Todo medidor desenha em duas camadas. A camada estática contém o mostrador, as escalas, as marcas, os rótulos e as faixas, e só é desenhada novamente quando a configuração muda. A camada dinâmica contém os ponteiros e o texto do valor. Um novo valor redesenha apenas a camada dinâmica, de modo que um painel com muitos medidores continua rápido enquanto os valores mudam.
sgcGauges faz parte da suíte visual da eSeGeCe. É desenhado com sgcUICore, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gerador de SVG e os temas, de modo que os medidores combinam com os outros produtos da suíte. sgcGauges requer sgcUICore, que é instalado junto com ele. A ponte sgcGaugesWS da edição Registered alimenta os medidores com mensagens MQTT, WebSocket e eventos enviados pelo servidor, por meio do sgcWebSockets.

Edições

sgcGauges é distribuído como Registered, Trial e Community. A Community mantém os medidores radial, linear, digital (7 segmentos), LED, barra de LEDs e anel de progresso, com faixas, limiares, animação, temas e a saída PNG e SVG sem janela, para VCL e FireMonkey. Deixa de fora o VU meter, o relógio, a bússola e o anunciador, as escalas e os ponteiros além do primeiro, os displays de 14 segmentos e de matriz de pontos e a ponte. Veja Edições.

Por onde começar

- Arquitetura: os modelos, os hosts, as duas camadas de renderização e as threads.
- Tipos de medidor: os dez medidores, suas classes e units num relance.
- Instalação e Plataformas: o instalador, os pacotes, as versões do Delphi e do C++Builder e as plataformas.
- Início rápido VCL, Início rápido FMX, Início rápido sem janela e Início rápido C++Builder: um primeiro medidor em poucos minutos.
- Componentes: uma página por controle, do medidor radial ao anunciador, e as classes de modelo TsgcCustomGauge, TsgcGaugeScale e TsgcGaugePointer.
- Escalas, Faixas, Ponteiros, Limiares, Animação, Temas, Saída sem janela e Edições.
- Ponte: alimente medidores com mensagens MQTT, WebSocket e eventos enviados pelo servidor.
- Solução de problemas: problemas comuns e suas causas.

Arquitetura

sgcGauges separa o que um medidor é do lugar onde ele é mostrado. Uma classe de modelo guarda a configuração e os valores e se desenha em um canvas do sgcUICore. Um controle host coloca um modelo em um formulário VCL ou FireMonkey.

Modelos

Todo tipo de medidor é uma classe que descende de **TsgcCustomGauge** (unit sgcGauges_Model), como TsgcRadialGauge ou TsgcLinearGauge. Um modelo é um TPersistent sem janela, de modo que também roda em um programa de console, em um serviço ou no Linux64. Ele contém:

- **Scales**: uma coleção de TsgcGaugeScale, cada uma com uma faixa de valores, marcas, rótulos e uma coleção de **Ranges** de cor.
- **Pointers**: uma coleção de TsgcGaugePointer, cada um mostrando um valor em uma escala.
- **Thresholds**: uma coleção de TsgcGaugeThreshold, valores que disparam OnThreshold quando um ponteiro os cruza.
- **Animation**: a duração e a suavização de um movimento de ponteiro.
- A aparência compartilhada: Theme, Title, TitleFont, Units, ShowValue, ValueFormat, ValueFont, BackgroundColor e FaceColor.

Um novo modelo sempre tem uma escala, de 0 a 100, e um ponteiro. **Value** é um atalho para Pointers[0].Value, e SetValue(aPointer, aValue, aAnimate) escreve o valor de qualquer ponteiro.

Hosts

Os controles VCL descendem de **TsgcVCLCustomGauge** (unit sgcGauges_VCL) e os controles FireMonkey de **TsgcFMXCustomGauge** (unit sgcGauges_FMX). Cada controle cria o modelo do seu tipo, que a propriedade pública **Gauge** retorna, e publica as propriedades do modelo que seu tipo desenha. Um host se repinta quando o modelo muda, executa os quadros de animação, entrega OnThreshold na thread principal e se salva com SaveToPNG e SaveToSVG no seu tamanho de cliente.

Duas camadas de renderização

Um modelo desenha em duas camadas:

- A **camada estática** (RenderStatic): o fundo, o mostrador, o aro, as escalas, as marcas, os rótulos, as faixas, as marcas de limiar e o título. Uma mudança de propriedade dispara OnChange com gckStatic e incrementa StaticStamp.
- A **camada dinâmica** (RenderDynamic): os ponteiros e o texto do valor. Um novo valor dispara OnChange apenas com gckDynamic.

Os hosts mantêm a camada estática como pixels em cache e desenham apenas a camada dinâmica enquanto um valor muda ou um ponteiro se move. BeginUpdate e EndUpdate reúnem várias mudanças em um único OnChange, estático quando alguma das mudanças foi estática. Render desenha as duas camadas de uma vez.

Tamanhos, ângulos e cores

- Os tamanhos das escalas e dos ponteiros (Radius, Offset, Thickness, Width, Length) são frações do tamanho do medidor, de modo que um medidor tem o mesmo aspecto em qualquer tamanho. O texto também se adapta ao medidor.
- Os ângulos são em graus, 0 são as 3 horas e os ângulos positivos giram no sentido horário. O relógio e a bússola contam a partir das 12 horas e do norte.
- As cores são valores TsgcUIColor na forma $AARRGGBB. Uma cor igual a 0 usa a cor do tema.
- Os valores são limitados à faixa de sua escala quando são escritos. NaN é ignorado e um valor infinito vai para o limite mais próximo.

Threads

Value, SetValue e o Value e o EndValue de um ponteiro podem ser escritos a partir de qualquer thread. Em um modelo, OnChange e OnThreshold são então chamados nessa thread. Os controles VCL e FireMonkey chamam seu OnThreshold na thread principal, seja qual for a thread que escreveu o valor. Todas as outras propriedades são destinadas à thread principal.

Units

Unit	Conteúdo
`sgcGauges_Types`	As enumerações e as funções auxiliares.
`sgcGauges_Model`	TsgcCustomGauge, as escalas, faixas, ponteiros, limiares e a animação.
`sgcGauges_Render`	O painter com o qual todo tipo de medidor desenha.
`sgcGauges_Radial`, `_Linear`, `_Digital`, `_LED`, `_ProgressRing`, `_VUMeter`, `_Clock`, `_Compass`, `_Annunciator`	As classes de modelo dos dez tipos de medidor. O LED e a barra de LEDs compartilham sgcGauges_LED.
`sgcGauges_VCL`	Os controles VCL.
`sgcGauges_FMX`	Os controles FireMonkey.
`sgcGaugesWS_Source`, `_Source_MQTT`, `_Source_WebSocket`, `_Source_SSE`	A ponte sgcGaugesWS, somente na edição Registered.

Veja também

- Tipos de medidor
- Saída sem janela
- Animação

Tipos de medidor

sgcGauges tem dez tipos de medidor. Cada um é uma classe de modelo em sua própria unit, um controle VCL em sgcGauges_VCL e um controle FireMonkey em sgcGauges_FMX. Os controles estão na página **SGC Gauges** da paleta.

Medidor	Classe de modelo (unit)	Controles VCL e FireMonkey	Edição	O que mostra
Radial	TsgcRadialGauge (sgcGauges_Radial)	TsgcVCLRadialGauge, TsgcFMXRadialGauge	Community	Um mostrador com arco completo, meio, quarto ou personalizado, um aro e o texto do valor. RadialStyle, StartAngle, SweepAngle, BezelStyle, ValuePosition.
Linear	TsgcLinearGauge (sgcGauges_Linear)	TsgcVCLLinearGauge, TsgcFMXLinearGauge	Community	Uma barra horizontal ou vertical, um termômetro ou um gráfico bullet com uma linha de alvo. Orientation, LinearStyle, Target, BulbSize.
Digital	TsgcDigitalGauge (sgcGauges_Digital)	TsgcVCLDigitalGauge, TsgcFMXDigitalGauge	Community (7 segmentos)	Um display de segmentos de um valor ou de um texto: 7 segmentos, 14 segmentos ou uma matriz de pontos 5 x 7. DisplayType, Digits, Text, Skew, SegmentColor, GhostColor, ShowGhost, Alignment.
LED	TsgcLEDGauge (sgcGauges_LED)	TsgcVCLLEDGauge, TsgcFMXLEDGauge	Community	Uma lâmpada acesa ou apagada, com brilho e piscada opcional. Active, OnColor, OffColor, Shape, Blink, BlinkInterval.
Barra de LEDs	TsgcLEDBarGauge (sgcGauges_LED)	TsgcVCLLEDBarGauge, TsgcFMXLEDBarGauge	Community	Uma linha ou coluna de segmentos de LED com retenção de pico. Segments, SegmentGap, Orientation, PeakHold, PeakHoldTime.
Anel de progresso	TsgcProgressRingGauge (sgcGauges_ProgressRing)	TsgcVCLProgressRingGauge, TsgcFMXProgressRingGauge	Community	Um anel que se preenche com o valor, com uma porcentagem no meio. Thickness, TrackColor, RoundCaps, ShowPercent.
VU meter	TsgcVUMeterGauge (sgcGauges_VUMeter)	TsgcVCLVUMeterGauge, TsgcFMXVUMeterGauge	Registered	Um medidor de nível de áudio com uma agulha ou uma barra e retenção de pico.
Relógio	TsgcClockGauge (sgcGauges_Clock)	TsgcVCLClockGauge, TsgcFMXClockGauge	Registered	Um relógio analógico, ao vivo ou em uma hora determinada, com diferença UTC opcional.
Bússola	TsgcCompassGauge (sgcGauges_Compass)	TsgcVCLCompassGauge,	Registered	Uma bússola com rosa giratória ou agulha giratória e um rumo de alvo.

		TsgcFMXCompassGauge		
Anunciador	TsgcAnnunciatorGauge (sgcGauges_Annunciator)	TsgcVCLAnnunciatorGauge, TsgcFMXAnnunciatorGauge	Registered	Uma grade de blocos de alarme em quatro estados, que piscam até serem reconhecidos.

O que todo medidor compartilha

Todo tipo descende de TsgcCustomGauge e, portanto, tem Scales, Pointers, Thresholds, Animation, Theme e as propriedades do texto do valor, mas cada controle publica apenas o que seu tipo desenha. O LED, por exemplo, não tem escala, o relógio não tem Value e a bússola publica Heading em vez de Value. Units é desenhado ao lado do texto do valor, e o texto do valor usa ValueFormat ou, quando está vazio, as casas decimais da escala.

Veja também

- Arquitetura
- Edições
- Início rápido VCL

Instalação

sgcGauges faz parte da distribuição sgcUI. Um único instalador cobre os produtos sgcUI e instala apenas os produtos que a sua licença inclui.

Instalação com o instalador sgcUI

1. Execute o instalador sgcUI e entre com o nome de usuário e a senha da sua conta eSeGeCe.
2. Escolha as versões do Delphi e do C++Builder nas quais instalar.
3. Na página de componentes, selecione **sgcGauges** e os outros produtos que desejar. O servidor de licenças retorna os produtos que você possui, e o instalador mostra uma caixa de seleção por produto. Os produtos que sua licença não inclui aparecem esmaecidos. Uma licença All-Access desbloqueia todos os produtos.
4. O instalador copia os arquivos, depois compila os pacotes na sua máquina para cada IDE selecionada, em ordem de dependência, e instala os pacotes de tempo de projeto.

O que acompanha o sgcGauges

sgcUICore, a base compartilhada dos produtos visuais sgcUI, acompanha o sgcGauges. Ele não tem uma caixa de seleção própria. O instalador o instala automaticamente sempre que sgcGauges é selecionado.

Pastas instaladas

Cada produto é instalado em uma pasta, por padrão `Documents\esegece\sgcUI` (`sgcUI_Trial` para o trial e `sgcUI_Community` para a edição Community).

Pasta	Conteúdo
`Source`	O código-fonte de todos os produtos instalados. Este é o único caminho de biblioteca do sgcUI.
`Packages`	Os pacotes de todos os produtos instalados, um conjunto por produto e versão do IDE.
`Lib<V>`	As units compiladas para cada versão do IDE e plataforma, por exemplo `LibD13`.
`Demos\Gauges`	Os demos do sgcGauges: Gallery, GalleryFMX, Headless, HMIPanel, DashboardGauges e MQTTLive, para Delphi e C++Builder.

O instalador adiciona `sgcUI\Source` ao caminho de biblioteca e ao caminho de navegação de cada IDE e plataforma selecionados, junto com a pasta das units compiladas. sgcUICore fica na mesma pasta `Source`, de modo que não precisa de um caminho próprio.

A paleta de componentes

Os componentes estão na página **SGC Gauges** da paleta. Um formulário VCL mostra os controles VCL e um formulário FireMonkey os controles FMX. As edições Registered e Trial registram 20 controles, os dez tipos de medidor para cada framework, e a ponte acrescenta suas três fontes de alimentação (TsgcGaugeFeedSource_WebSocket, TsgcGaugeFeedSource_MQTT e TsgcGaugeFeedSource_SSE). A Community registra 12 controles, os seis tipos de medidor da Community para cada framework.

Trial e Community

- **Trial**: o instalador de avaliação do sgcUI instala todos os produtos em modo de avaliação, sgcGauges incluído.

- **Community**: o instalador gratuito sgcUI Community inclui o sgcGauges sem seus recursos avançados. Veja Edições.

Compilação manual dos pacotes

Para compilar os pacotes você mesmo, abra-os para a sua versão do IDE a partir de `sgcUI\Packages`. `<V>` é o sufixo da versão do Delphi, por exemplo `sgcGaugesD13.dpk`.

Pacote	Conteúdo
`sgcUICore<V>` , `sgcUICoreVCL<V>` , `sgcUICoreFMX<V>`	A base compartilhada sgcUICore e suas partes VCL e FireMonkey (runtime).
`sgcGauges<V>`	As units independentes de framework: os tipos, o modelo, o painter e os dez modelos de medidor (runtime).
`sgcGaugesVCL<V>` / `dclsgcGauges<V>`	Os controles VCL (runtime / tempo de projeto).
`sgcGaugesFMX<V>` / `dclsgcGaugesFMX<V>`	Os controles FireMonkey, a partir do Delphi XE6 (runtime / tempo de projeto).
`sgcGaugesWS<V>` / `dclsgcGaugesWS<V>`	A ponte para o sgcWebSockets, somente na edição Registered. Requer o pacote sgcWebSockets.

Compile-os em ordem de dependência: `sgcUICore`, `sgcUICoreVCL` e `sgcUICoreFMX`, depois `sgcGauges`, `sgcGaugesVCL` e `sgcGaugesFMX`, depois os pacotes de tempo de projeto. O grupo de projetos `sgcGauges<V>.groupproj` contém os pacotes do sgcGauges nessa ordem. Em seguida, instale os pacotes de tempo de projeto e adicione `sgcUI\Source` ao caminho de biblioteca da sua versão do Delphi (Tools > Options > Language > Delphi Options > Library > Library path). Para o C++Builder, abra os projetos `.cbproj` correspondentes à sua versão, por exemplo `sgcGaugesC13.cbproj` e `sgcGaugesVCLC13.cbproj`, e compile-os. Os cabeçalhos `.hpp` gerados aparecem ao lado do pacote compilado.

Veja também

- Plataformas
- Início rápido VCL

Plataformas

Destino	Suporte
Delphi	Delphi 7, 2007, 2009, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13.
C++Builder	C++Builder 2007, 2010, XE a XE8, 10 a 10.4, 11, 12 e 13, com os projetos de pacote sgcGaugesC2007 a sgcGaugesC13. Os pacotes geram os cabeçalhos .hpp e os arquivos .lib que um projeto C++Builder inclui e vincula. Veja Início rápido C++Builder.
VCL	Os controles TsgcVCL no Windows, Win32 e Win64, em todas as versões do Delphi.
FireMonkey	Os controles TsgcFMX a partir do Delphi XE6 (SGC_GAUGES_FMX), nas plataformas FireMonkey. Usam apenas FireMonkey e sgcUICore.
Linux64	As units independentes de framework a partir do Delphi 10.2 (SGC_GAUGES_LINUX): o modelo, o painter e os dez modelos de medidor, com a saída PNG e SVG sem janela. Não há controles para Linux.
.NET	Uma versão C# com a mesma API: os dez medidores como controles WinForms, WPF e Avalonia e a saída PNG e SVG sem janela, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão geral do .NET.

Pacotes

O sgcGauges traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcGaugesD13.dpk para o Delphi 13 e sgcGaugesC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcGauges	Runtime, independente de framework	Toda unidade do produto que não usa nem a VCL nem o FireMonkey: o modelo, o painter e os dez modelos de medidor, com a saída PNG e SVG sem janela.	rtl, sgcUICore
sgcGaugesVCL	Runtime, VCL	sgcGauges_VCL (TsgcVCLRadialGauge, TsgcVCLLinearGauge, TsgcVCLDigitalGauge, TsgcVCLLEDGauge, TsgcVCLLEDBarGauge, TsgcVCLProgressRingGauge, TsgcVCLVUMeterGauge, TsgcVCLClockGauge, TsgcVCLCompassGauge e TsgcVCLAnnunciatorGauge)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcGauges
sgcGaugesFMX	Runtime, FireMonkey, a partir do XE6	sgcGauges_FMX (TsgcFMXRadialGauge, TsgcFMXLinearGauge, TsgcFMXDigitalGauge, TsgcFMXLEDGauge, TsgcFMXLEDBarGauge, TsgcFMXProgressRingGauge, TsgcFMXVUMeterGauge, TsgcFMXClockGauge, TsgcFMXCompassGauge e TsgcFMXAnnunciatorGauge)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcGauges
dclsgcGauges	Tempo de projeto, VCL	Registra os medidores VCL na página **SGC Gauges** da paleta (TsgcVCLVUMeterGauge, TsgcVCLClockGauge, TsgcVCLCompassGauge e TsgcVCLAnnunciatorGauge não na edição Community), com o editor de coleções para Scales, Pointers, Thresholds e Tiles.	designide, rtl, vcl, sgcGauges, sgcGaugesVCL

dclsgcGaugesFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra os medidores FireMonkey na página **SGC Gauges** da paleta.	designide, rtl, fmx, sgcGauges, sgcGaugesFMX, dclsgcGauges
sgcGaugesWS, dclsgcGaugesWS	Runtime e tempo de projeto, ponte do sgcWebSockets	As fontes de dados em tempo real (TsgcGaugeFeedSource_WebSocket, TsgcGaugeFeedSource_MQTT e TsgcGaugeFeedSource_SSE), independentes de framework, na página **SGC Gauges** da paleta.	rtl, sgcGauges, sgcWebSockets

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcGaugesFMX exige o dclsgcGauges, portanto instale primeiro o dclsgcGauges. Nenhuma unit do sgcGauges usa uma unit do sgcWebSockets, exceto as units da ponte opcional sgcGaugesWS.

Versões do IDE e plataformas de destino

Delphi	sgcGauges	sgcGaugesVCL	sgcGaugesFMX	dclsgcGauges, dclsgcGaugesFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE. O sgcGaugesWS é compilado para as mesmas plataformas que o sgcGauges, e o dclsgcGaugesWS para as mesmas plataformas que o dclsgcGauges.

C++Builder	sgcGauges, sgcGaugesVCL	sgcGaugesFMX	dclsgcGauges, dclsgcGaugesFMX

2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcGauges)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcGauges)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcGaugesC13.cbproj, sgcGaugesVCLC13.cbproj, sgcGaugesFMXC13.cbproj, dclsgcGaugesC13.cbproj e dclsgcGaugesFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. O sgcGaugesWS e o dclsgcGaugesWS são compilados para as mesmas versões e plataformas do C++Builder que o sgcGauges e o dclsgcGauges, exceto Win64x, porque precisam de um build Win64x do pacote sgcWebSockets. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXRadialGauge, TsgcFMXLinearGauge, TsgcFMXDigitalGauge, TsgcFMXLEDGauge, TsgcFMXLEDBarGauge, TsgcFMXProgressRingGauge, TsgcFMXVUMeterGauge, TsgcFMXClockGauge, TsgcFMXCompassGauge e TsgcFMXAnnunciatorGauge) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcGaugesVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcGauges_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcGauges é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa o modelo, o painter e os dez modelos de medidor, com a saída PNG e SVG sem janela. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Restrições de versão

- As units independentes de framework e as units VCL compilam no Delphi 7: sem generics, sem métodos anônimos, sem variáveis inline e sem record helpers.
- A saída é a mesma no Win32, no Win64 e no Linux64: os arquivos PNG e SVG de um medidor são idênticos byte a byte nas três plataformas.
- As enumerações têm prefixos de medidor (grs, gpk, gdt, gtd, ge e assim por diante), de modo que os cabeçalhos do C++Builder nunca colidem com os outros produtos sgcUI.

Veja também

- Edições
- Arquitetura

Início rápido VCL

Instale os pacotes do sgcGauges para a sua versão do Delphi. Os componentes estão na página **SGC Gauges** da paleta.

Em tempo de projeto

1. Coloque um **TsgcVCLRadialGauge** em um formulário. Digite Speed em Title e km/h em Units.
2. Abra a coleção Scales, selecione a primeira escala e defina Maximum como 260 e MajorInterval como 20. Abra sua coleção Ranges, adicione uma faixa e defina StartValue como 200, EndValue como 260 e Color como um vermelho como $FFE34948.
3. Coloque um **TsgcVCLLEDGauge** ao lado dele, defina OnColor com o mesmo vermelho e Blink como True.
4. Adicione um limiar à coleção Thresholds do medidor radial com Value 200, Hysteresis 10 e Direction gtd-Both, e escreva um manipulador OnThreshold que atribua aRising à propriedade Active do LED.
5. Defina Value no código, a partir de um timer ou de uma worker thread, e execute a aplicação. A agulha se move para cada novo valor em 300 ms.

No código

Um medidor funciona assim que tem um Parent. Defina a faixa de uma escala antes do valor, porque um valor é limitado à escala quando é escrito.

```
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;
```

Veja também

- Tipos de medidor
- Limiares
- Animação

Início rápido FMX

Os controles FireMonkey têm as mesmas propriedades que os VCL, com o prefixo TsgcFMX, a partir do Delphi XE6. Instale os pacotes do sgcGauges, abra um formulário FireMonkey e use os componentes da página **SGC Gauges** da paleta.

Em tempo de projeto

1. Coloque um **TsgcFMXRadialGauge** em um formulário, defina RadialStyle como grsHalf e Theme como utDark.
2. Defina Title, Units e ValueFormat, e o Maximum da primeira escala na coleção Scales.
3. Coloque um **TsgcFMXLinearGauge**, defina Orientation como gloVertical e LinearStyle como glsThermometer.
4. Execute a aplicação em qualquer plataforma FireMonkey.

O que saber no FireMonkey

- Os controles desenham na escala da cena, de modo que ficam nítidos em telas de DPI alto e em dispositivos móveis.
- OnThreshold é chamado na thread principal por meio de TThread.Queue, seja qual for a thread que escreveu o valor.

No 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;
```

Veja também

- Início rápido VCL
- Temas
- Plataformas

Início rápido sem janela

As classes de modelo são renderizadas sem janela. Um programa de console, um servidor web ou um relatório pode criar um medidor, definir seus valores e salvá-lo como arquivo PNG ou SVG. As units do modelo não usam VCL nem FireMonkey, de modo que o mesmo código roda no Windows e no Linux64.

Passos

1. Adicione sgcGauges_Model e a unit do tipo de medidor à cláusula uses, por exemplo sgcGauges_Radial.
2. Crie o modelo, por exemplo `TsgcRadialGauge.Create`, e defina suas escalas, faixas e valor como em um controle.
3. Chame SaveToPNG ou SaveToSVG com o tamanho em pixels lógicos e uma escala opcional, depois libere o modelo.

No 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;
```

Uma renderização sem janela desenha os valores de destino. A animação e a temporização dos quadros pertencem aos controles. Veja Saída sem janela.

Veja também

- Saída sem janela
- Arquitetura

Início rápido C++Builder

Instale os pacotes do sgcGauges para a sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página **SGC Gauges** da paleta exatamente como no Delphi. O instalador adiciona os dois caminhos ao IDE para você. Os cabeçalhos .hpp e os arquivos .lib gerados do sgcGauges e do sgcUICore estão nessa única pasta.

O que saber no C++

- Inclua o cabeçalho da unit que declara um controle: `sgcGauges_VCL.hpp` para os controles VCL e `sgcGauges_FMX.hpp` para os FireMonkey. As coleções e os tipos de evento estão em `sgcGauges_Model.hpp` e as enumerações em `sgcGauges_Types.hpp`.
- As coleções não têm índice padrão no C++: escreva `Gauge->Scales->Items[0]` onde o Delphi escreve `Gauge.Scales[0]`.
- Os eventos são closures, de modo que seus manipuladores são funções membro __fastcall de uma classe.
- As cores são valores de 32 bits sem sinal na forma 0xAARRGGBB.
- As classes de modelo têm construtores sem argumentos, como `new TsgcRadialGauge()`, e você mesmo as exclui com delete. Veja Saída sem janela.

No código

Veja também

- Início rápido VCL
- Instalação

Medidor radial

Classes: TsgcVCLRadialGauge (unit sgcGauges_VCL), TsgcFMXRadialGauge (unit sgcGauges_FMX) e o modelo TsgcRadialGauge (unit sgcGauges_Radial)
Um mostrador: as escalas sobre um arco, um aro ao redor da face, os ponteiros e o texto do valor. O arco e seu tamanho se ajustam ao controle, de modo que um meio medidor fica com a parte inferior reta e preenche um controle largo.
O medidor faz parte de todas as edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
RadialStyle	TsgcGaugeRadialStyle	O arco: grsFull (padrão, a partir de 135 graus ao longo de 270), grsHalf (180 ao longo de 180), grsQuarter (180 ao longo de 90) ou grsCustom.
StartAngle	Double	O início do arco com grsCustom, em graus, 0 são as 3 horas e positivo é no sentido horário. Padrão 135.
SweepAngle	Double	O comprimento do arco com grsCustom, em graus. Padrão 270.
BezelStyle	TsgcGaugeBezelStyle	gbzNone, gbzThin (padrão) ou gbzThick.
ValuePosition	TsgcGaugeValuePosition	Onde vai o texto do valor: gvpBottom (padrão), gvpCenter ou gvpTop.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Os valores que disparam OnThreshold quando um ponteiro os cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
Units	string	O texto desenhado ao lado do texto do valor, como km/h.
ShowValue	Boolean	Desenha o texto do valor. Padrão True.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais da escala.
ValueFont	TsgcUITextStyle	A fonte do texto do valor.

BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
FaceColor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) substitui o texto do valor.

Exemplo

Um meio tacômetro com uma faixa vermelha:

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

Medidor linear

Classes: TsgcVCLLinearGauge (unit sgcGauges_VCL), TsgcFMXLinearGauge (unit sgcGauges_FMX) e o modelo TsgcLinearGauge (unit sgcGauges_Linear)
Um medidor horizontal ou vertical desenhado como barra, termômetro ou gráfico bullet. O ponteiro padrão é uma barra.
O medidor faz parte de todas as edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Orientation	TsgcGaugeOrientation	gloHorizontal (padrão) vai da esquerda para a direita, gloVertical de baixo para cima.
LinearStyle	TsgcGaugeLinearStyle	glsBar (padrão): a trilha da escala e os ponteiros. glsThermometer: a escala 0 é um tubo com um bulbo no mínimo, e o ponteiro 0 é desenhado como o líquido. glsBullet: faixas largas, as faixas de cor atrás de uma barra mais fina e uma linha Target.
Target	Double	A linha de alvo do estilo bullet, desenhada através da escala 0 quando está dentro da escala. Padrão 0.
BulbSize	Double	O raio do bulbo do termômetro como fração do tamanho do medidor. Padrão 0.06.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Os valores que disparam OnThreshold quando um ponteiro os cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
Units	string	O texto desenhado ao lado do texto do valor, como km/h.
ShowValue	Boolean	Desenha o texto do valor. Padrão True.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais da escala.

ValueFont	TsgcUI-TextStyle	A fonte do texto do valor.
Back-ground-Color	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
FaceCo-lor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é ar-mazenado no formulário.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) substitui o texto do valor.

Exemplo

Um termômetro vertical e um 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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

Medidor digital

Classes: TsgcVCLDigitalGauge (unit sgcGauges_VCL), TsgcFMXDigitalGauge (unit sgcGauges_FMX) e o modelo TsgcDigitalGauge (unit sgcGauges_Digital)
Um display de segmentos para o valor do ponteiro 0, ou para um texto. O tipo de 7 segmentos mostra dígitos, o sinal de menos, o ponto decimal e os dois pontos; o tipo de 14 segmentos acrescenta letras; a matriz de pontos 5 x 7 mostra ASCII 32 a 126. Os segmentos apagados podem ser desenhados com pouca intensidade atrás dos acesos.
O número de posições de glifo vem de Text, ou do maior entre os textos do Minimum e do Maximum da escala, a menos que Digits o defina. Um valor que não cabe mostra traços, e um Text que não cabe é cortado após Digits glifos. Nos tipos de segmentos, um '.' ou ',' acende o ponto decimal do glifo anterior e não ocupa posição própria.
O display de 7 segmentos faz parte de todas as edições; na Community os tipos de 14 segmentos e de matriz de pontos são desenhados como 7 segmentos. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
DisplayType	TsgcGaugeDisplayType	gdt7Segment (padrão), gdt14Segment ou gdtDotMatrix.
Digits	Integer	O número de posições de glifo. 0 (padrão) toma a contagem do texto.
Text	string	Um texto mostrado no lugar do valor. Vazio (padrão) mostra o valor.
Skew	Double	Inclina os glifos, em graus de -30 a 30. Positivo move o topo para a direita. Padrão 0.
SegmentColor	TsgcUIColor	A cor dos segmentos acesos. 0 usa a cor de ValueFont, senão a cor de texto do tema. O ValueColor de um limiar cruzado tem prioridade sobre ela.
GhostColor	TsgcUIColor	A cor dos segmentos apagados. 0 usa a cor dos acesos, esmaecida.
ShowGhost	Boolean	Desenha os segmentos apagados. Padrão True.
Alignment	TsgcUITextHAlign	A posição de um texto mais curto. Padrão thaRight.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Os valores que disparam OnThreshold quando um ponteiro os cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.

TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
Units	string	O texto desenhado ao lado do texto do valor, como km/h.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais da escala.
ValueFont	TsgcUITextStyle	A fonte do texto do valor.
BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
FaceColor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) substitui o texto do valor.

Exemplo

Um contador de seis dígitos em verde e um texto fixo:

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

LED

Classes: TsgcVCLLEDGauge (unit sgcGauges_VCL), TsgcFMXLEDGauge (unit sgcGauges_FMX) e o modelo TsgcLEDGauge (unit sgcGauges_LED)
Uma lâmpada indicadora com brilho, que pode piscar. A lâmpada fica acesa enquanto Active é True. Active é uma visão de Value <> 0, de modo que um vínculo de alimentação que escreve 0 ou 1 no valor liga e desliga a lâmpada. O LED guarda qualquer valor exceto NaN, sem limitar a uma escala, e nunca anima.
O medidor faz parte de todas as edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Active	Boolean	Acende a lâmpada. Atribuí-la escreve 1 ou 0 no valor. Não é armazenado no formulário.
OnColor	TsgcUIColor	A cor da lâmpada acesa. 0 é o verde da paleta do tema.
OffColor	TsgcUIColor	A cor da lâmpada apagada. 0 usa a cor da acesa, translúcida.
Shape	TsgcGaugeLEDShape	gshCircle (padrão), gshSquare ou gshRect.
Blink	Boolean	A lâmpada acesa liga e desliga a cada BlinkInterval. Padrão False. Não há piscada em tempo de projeto.
BlinkInterval	Integer	O período de piscada em milissegundos. Padrão 500, no mínimo 50.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.

Exemplo

Uma lâmpada de alarme vermelha que pisca enquanto está ligada:

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge
- Barra de LEDs

Barra de LEDs

Classes: TsgcVCLLEDBarGauge (unit sgcGauges_VCL), TsgcFMXLEDBarGauge (unit sgcGauges_FMX) e o modelo TsgcLEDBarGauge (unit sgcGauges_LED)
Uma linha ou uma coluna de segmentos de LED que dividem a escala do ponteiro 0. Um segmento fica aceso quando seu centro está entre a origem do ponteiro e o seu valor. Um segmento assume a cor da faixa da escala em que está, senão verde, âmbar e vermelho por terços da escala. A retenção de pico mantém o segmento mais alto aceso por um tempo depois que o nível cai.
O medidor faz parte de todas as edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Segments	Integer	O número de segmentos, de 1 a 100. Padrão 10.
SegmentGap	Double	O espaço entre dois segmentos como fração de um segmento, até 0.9. Padrão 0.2.
Orientation	TsgcGaugeOrientation	gloHorizontal (padrão) da esquerda para a direita, gloVertical de baixo para cima.
PeakHold	Boolean	Mantém o segmento de pico aceso. Padrão False.
PeakHoldTime	Integer	Por quanto tempo o pico é retido, em milissegundos. Padrão 1000.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.

Exemplo

Uma barra de nível vertical de 20 segmentos com uma faixa azul e retenção 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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge
- Faixas

Anel de progresso

Classes: TsgcVCLProgressRingGauge (unit sgcGauges_VCL), TsgcFMXProgressRingGauge (unit sgcGauges_FMX) e o modelo TsgcProgressRingGauge (unit sgcGauges_ProgressRing)
Um anel que se preenche no sentido horário a partir das 12 horas com o valor do ponteiro 0, com a porcentagem no meio. A barra assume o Color do ponteiro, senão a cor da faixa que contém o valor, senão a cor de destaque do tema. O ponteiro padrão é uma barra.
O medidor faz parte de todas as edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Thickness	Double	A largura do anel como fração do tamanho do medidor. Padrão 0.15.
TrackColor	TsgcUIColor	A cor do anel vazio. 0 usa o tema.
RoundCaps	Boolean	Arredonda as duas pontas de uma barra mais curta que o anel completo. Padrão True.
ShowPercent	Boolean	True (padrão) mostra a porcentagem do valor dentro da escala, com ValueFormat ou '0' e um sinal de %. False mostra o texto do valor e Units.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Os valores que disparam OnThreshold quando um ponteiro os cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
Units	string	O texto desenhado ao lado do texto do valor, como km/h.
ShowValue	Boolean	Desenha o texto do valor. Padrão True.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais da escala.
ValueFont	TsgcUITextStyle	A fonte do texto do valor.

BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) substitui o texto do valor quando ShowPercent é False.

Exemplo

O progresso de um upload, com a contagem 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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

VU meter

Classes: TsgcVCLVUMeterGauge (unit sgcGauges_VCL), TsgcFMXVUMeterGauge (unit sgcGauges_FMX) e o modelo TsgcVUMeterGauge (unit sgcGauges_VUMeter)
Um medidor de nível de áudio. A escala padrão vai de -20 a +3 dB com o mapeamento em decibéis, de modo que uma posição segue a amplitude, com as marcas por fora e uma faixa vermelha a partir de 0 dB. O ponteiro se move em amplitude, como um medidor real. Enquanto MajorInterval é 0, a escala mostra as marcas clássicas do VU.
O medidor precisa da edição Registered ou Trial: na Community o modelo não desenha nada e os controles não são instalados. Veja Edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
MeterStyle	TsgcGaugeVUMeterStyle	gvmNeedle (padrão): uma agulha em um arco de 90 graus. gvmBar: uma barra horizontal sobre uma faixa com a trilha e as faixas de cor.
PeakHold	Boolean	Mostra o pico do nível. Padrão False.
PeakHoldTime	Integer	Por quanto tempo o pico é retido antes de cair, em milissegundos. Padrão 1500.
PeakDecay	Double	A velocidade com que o pico cai após a retenção, em dB por segundo. Padrão 20. 0 descarta o pico de uma vez após a retenção.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Thresholds	TsgcGaugeThresholds	Os valores que disparam OnThreshold quando um ponteiro os cruza.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
Units	string	O texto desenhado ao lado do texto do valor, como km/h.
ShowValue	Boolean	Desenha o texto do valor. Padrão True.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais da escala.
ValueFont	TsgcUITextStyle	A fonte do texto do valor.

Back-groundCo-lor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
FaceColor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.
Value	Double	O valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.

Métodos

ResetPeak limpa o pico.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) substitui o texto do valor.

Exemplo

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

Relógio

Classes: TsgcVCLClockGauge (unit sgcGauges_VCL), TsgcFMXClockGauge (unit sgcGauges_FMX) e o modelo TsgcClockGauge (unit sgcGauges_Clock)
Um relógio analógico com 60 marcas de minuto, 12 numerais e os ponteiros de hora, minuto e segundo. Mostra Time ou, com Live, a hora real: a hora local, ou a hora UTC mais UTCOffset minutos com UseUTCOffset. Um relógio ao vivo se repinta a cada segundo, ou a cada minuto sem ShowSeconds. Os numerais usam o LabelFont da escala 0. O relógio não tem Value nem ponteiros de valor.
O medidor precisa da edição Registered ou Trial: na Community o modelo não desenha nada e os controles não são instalados. Veja Edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Time	TDateTime	A hora mostrada enquanto Live é False.
Live	Boolean	Mostra a hora real. Padrão False.
UseUTCOffset	Boolean	Um relógio ao vivo mostra a hora UTC mais UTCOffset em vez da hora local. Padrão False.
UTCOffset	Integer	A diferença em relação ao UTC em minutos, como 60 ou -300. Padrão 0.
Numerals	TsgcGaugeClockNumerals	gcnArabic (padrão), gcnRoman ou gcnNone.
ShowSeconds	Boolean	Desenha o ponteiro dos segundos. Padrão True.
SmoothSeconds	Boolean	O ponteiro dos segundos desliza em vez de saltar. Padrão False.
HourHandColor	TsgcUIColor	A cor do ponteiro das horas. 0 usa o tema.
MinuteHandColor	TsgcUIColor	A cor do ponteiro dos minutos. 0 usa o tema.
SecondHandColor	TsgcUIColor	A cor do ponteiro dos segundos. 0 usa o tema.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
ValueFont	TsgcUITextStyle	A fonte do texto do valor.
BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.

FaceColor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.

O modelo também tem uma propriedade pública TimeSource, uma função que retorna a hora real. Quando é atribuída, um relógio ao vivo a lê em vez do relógio do sistema, o que dá quadros fixos nos testes.

Exemplo

Um relógio ao vivo para UTC+5:30 com numerais romanos, e uma hora fixa:

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

Bússola

Classes: TsgcVCLCompassGauge (unit sgcGauges_VCL), TsgcFMXCompassGauge (unit sgcGauges_FMX) e o modelo TsgcCompassGauge (unit sgcGauges_Compass)
Um indicador de rumo. Heading é o valor do ponteiro 0 em uma escala de 0 a 360 graus, no sentido horário a partir do norte. Ele dá a volta em vez de ser limitado (450 é 90, -30 é 330), e um movimento é animado pelo arco mais curto, de modo que de 359 para 1 gira 2 graus. A rosa tem uma marca a cada 10 graus, os pontos cardeais e rótulos de graus a cada 30 graus, no LabelFont da escala 0. O texto do rumo mostra graus inteiros e um sinal de grau, a menos que ValueFormat esteja definido.
O medidor precisa da edição Registered ou Trial: na Community o modelo não desenha nada e os controles não são instalados. Veja Edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Heading	Double	O rumo em graus, o valor do ponteiro 0. Pode ser escrito a partir de qualquer thread. Não é armazenado no formulário.
CompassStyle	TsgcGaugeCompassStyle	gcsRotatingCard (padrão): a rosa gira e o rumo fica sob uma marca fixa no topo. gcsRotatingNeedle: a rosa é fixa e uma agulha aponta para o rumo.
ShowCardinals	Boolean	Desenha N, E, S, W e os pontos intercardeais. Padrão True.
ShowDegrees	Boolean	Desenha os rótulos de graus da rosa. Padrão True. O texto do rumo segue ShowValue.
TargetHeading	Double	O rumo da marca de alvo. Padrão 0.
ShowTarget	Boolean	Desenha a marca de alvo na borda, na cor de destaque do tema. Padrão False.
Scales	TsgcGaugeScales	As escalas com suas marcas, rótulos e faixas de cor. Veja TsgcGaugeScale.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer.
Animation	TsgcGaugeAnimation	Enabled, Duration e Easing de um movimento de ponteiro.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
ShowValue	Boolean	Desenha o texto do valor. Padrão True.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais da escala.

ValueFont	TsgcUITextStyle	A fonte do texto do valor.
BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
FaceColor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.

Eventos

OnFormatValue(Sender, aPointer, aValue, var aText) substitui o texto do rumo.

Exemplo

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

Anunciador

Classes: TsgcVCLAnnunciatorGauge (unit sgcGauges_VCL), TsgcFMXAnnunciatorGauge (unit sgcGauges_FMX) e o modelo TsgcAnnunciatorGauge (unit sgcGauges_Annunciator)
Um painel de alarmes: uma grade de blocos, preenchida linha por linha abaixo do título. Cada bloco tem um Caption, um State (gasNormal, gasAdvisory, gasWarning ou gasAlarm) e um sinalizador Acknowledged, e assume a cor do seu estado. Um aviso ou um alarme não reconhecido pisca; um aviso informativo fica fixo. Uma mudança de State limpa Acknowledged, de modo que cada novo estado precisa do seu próprio reconhecimento. Os rótulos usam o tamanho e o peso de ValueFont e são cortados na largura do bloco. O anunciador não tem escalas, ponteiros nem limiares.
O medidor precisa da edição Registered ou Trial: na Community o modelo não desenha nada e os controles não são instalados. Veja Edições. A classe VCL descende de TsgcVCLCustomGauge e a classe FireMonkey de TsgcFMXCustomGauge. Ambas publicam as propriedades da tabela abaixo, Theme, OnThreshold e MaxFPS (o limite de taxa de quadros da animação, padrão 60), além das habituais do seu framework, como Align, Anchors, Enabled, Visible, PopupMenu, OnClick e os eventos de mouse. A propriedade pública Gauge retorna o modelo, e SaveToPNG e SaveToSVG salvam o controle no seu tamanho de cliente vezes aScale.

Propriedades

Propriedade	Tipo	Descrição
Tiles	TsgcGaugeAnnunciatorTiles	Os blocos, itens de TsgcGaugeAnnunciatorTile com Caption, State e Acknowledged.
Columns	Integer	Os blocos por linha. Padrão 4.
FlashInterval	Integer	O período de piscada em milissegundos. Padrão 500, no mínimo 50.
NormalColor	TsgcUIColor	A cor de um bloco normal. 0 usa a cor do mostrador do tema.
AdvisoryColor	TsgcUIColor	A cor de um bloco informativo. 0 é azul.
WarningColor	TsgcUIColor	A cor de um bloco de aviso. 0 é âmbar.
AlarmColor	TsgcUIColor	A cor de um bloco de alarme. 0 é vermelho.
Title	string	O título do medidor.
TitleFont	TsgcUITextStyle	A fonte do título. Um Color igual a 0 segue o tema.
ValueFont	TsgcUITextStyle	A fonte do texto do valor.
BackgroundColor	TsgcUIColor	A cor atrás do medidor. 0 usa o fundo do tema.
FaceColor	TsgcUIColor	A cor do mostrador. 0 usa a do tema.

Métodos

Método	Descrição
Acknowledge(aIndex)	Reconhece um bloco que não está normal: ele para de piscar.

AcknowledgeAll	Reconhece todos os blocos que não estão normais.
TileAt(X, Y)	O índice do bloco em um ponto do controle, ou -1.
HoverTile	O índice do bloco sob o mouse, ou -1.

Eventos

OnTileClick(Sender, aIndex) do tipo TsgcGaugeTileClickEvent (unit sgcGauges_Annunciator) dispara quando o botão esquerdo é solto sobre um bloco.

Exemplo

Três blocos; um clique reconhece um bloco:

```
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;
```

Veja também

- Tipos de medidor
- TsgcCustomGauge

TsgcCustomGauge

Unit: sgcGauges_Model
A classe base dos dez modelos de medidor, um TPersistent sem janela. Um controle host cria o modelo do seu tipo e o retorna na sua propriedade Gauge; um programa sem interface de usuário cria o modelo ele mesmo, como TsgcRadialGauge.Create, e o libera. Um novo modelo tem uma escala de 0 a 100 e um ponteiro.

Propriedades

Propriedade	Tipo	Descrição
Scales	TsgcGaugeScales	As escalas. Veja TsgcGaugeScale. A Community usa apenas a primeira.
Pointers	TsgcGaugePointers	Os ponteiros. Veja TsgcGaugePointer. A Community usa apenas o primeiro.
Thresholds	TsgcGaugeThresholds	Os limiares. Veja TsgcGaugeThreshold.
Animation	TsgcGaugeAnimation	Enabled (padrão True), Duration em milissegundos (padrão 300) e Easing (padrão geCubicOut).
Theme	TsgcUITheme	utLight (padrão) ou utDark.
Title, TitleFont	string, TsgcUITextStyle	O título do medidor e sua fonte.
Units	string	O texto desenhado ao lado do texto do valor. ValueToText não o acrescenta.
ShowValue	Boolean	Desenha o texto do valor. Padrão True.
ValueFormat	string	A máscara FormatFloat do texto do valor. Vazia usa as casas decimais do passo principal da escala.
ValueFont	TsgcUITextStyle	A fonte do texto do valor. Um Color igual a 0 segue o tema.
BackgroundColor, FaceColor	TsgcUIColor	A cor atrás do medidor e a cor do mostrador. 0 usa o tema.
Value	Double	O valor do ponteiro 0, o mesmo que SetValue(0, Value).
FormatSettings	TFormatSettings	Pública. As configurações do texto do valor e dos rótulos, com '.' como separador decimal por padrão.
StaticStamp	Integer	Pública, somente leitura. Incrementada a cada mudança estática; um host a compara para saber se sua camada estática em cache está atual.
Clock	TsgcGaugeClockFunc	Pública. Uma função que retorna a hora em milissegundos para a animação. nil usa o relógio do sistema.
Owner	TPersistent	Pública. O controle host, ou nil para um modelo sem janela.
Designing	Boolean	Pública. Definida pelos hosts em tempo de projeto, onde nada anima.

Métodos

Método	Descrição
SetValue(aPointer, aValue, aAnimate = True)	Escreve o valor de um ponteiro, com uma animação quando aAnimate e Animation.Enabled são True. Thread safe.
BeginUpdate, EndUpdate	Reúnem as mudanças entre eles em um único OnChange.
DisplayValue(aPointer, aNowMs)	O valor desenhado de um ponteiro em um instante do Clock, durante uma animação.
ValueToText(aPointer, aValue)	O texto do valor de um valor, com ValueFormat e OnFormatValue.
IsAnimating(aNowMs), NextFrameMs(aNowMs)	Se um ponteiro se move e quando o próximo quadro temporizado (uma piscada, um tique do relógio) é devido.
ActiveScaleCount, ActivePointerCount, PointerScale(aPointer)	As escalas e os ponteiros que a edição desenha, e a escala de um ponteiro.
Render, RenderStatic, RenderDynamic	Desenham as duas camadas, ou uma, em um TsgcUICanvas de aWidth x aHeight pixels lógicos em aScale.
SaveToPNG, SaveToSVG	Salvam o medidor em um stream ou em um arquivo com aWidth x aHeight pixels lógicos, em aScale.

Eventos

Evento	Descrição
OnChange(Sender, aKind)	Uma mudança: gckStatic para a camada estática, gckDynamic para um valor ou um movimento de ponteiro. Chamado na thread da mudança.
OnThreshold(Sender, aThreshold, aValue, aRising)	Um ponteiro cruzou um limiar. Em um modelo é chamado na thread que escreveu o valor; os controles o chamam na thread principal.
OnFormatValue(Sender, aPointer, aValue, var aText)	Substitui o texto do valor.

Exemplo

Um medidor sem janela com dois ponteiros que conta suas mudanças:

```
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;
```

Veja também

- Arquitetura
- Saída sem janela
- Animação

TsgcGaugeScale e TsgcGaugeRange

Unit: sgcGauges_Model
TsgcGaugeScale é um item da coleção Scales (TsgcGaugeScales) de um medidor. Uma escala mapeia valores para posições no arco ou no eixo e desenha sua trilha, marcas, rótulos e faixas de cor. TsgcGaugeRange é um item da coleção Ranges (TsgcGaugeRanges) de uma escala. Os tamanhos são frações do tamanho do medidor.

TsgcGaugeScale

Propriedade	Tipo	Descrição
Minimum, Maximum	Double	A faixa de valores. Padrão 0 e 100. Defina-os antes dos valores, que são limitados ao serem escritos.
Mapping	TsgcGaugeScaleMapping	gsmLinear (padrão) ou gsmDecibel, em que uma posição segue a amplitude.
Reversed	Boolean	Percorre a escala de Maximum para Minimum. Padrão False.
Radius	Double	O raio externo da escala em um mostrador. Padrão 1.
Offset	Double	A posição da banda ao longo do eixo de um medidor linear. Padrão 0.
Thickness	Double	A largura da trilha. Padrão 0.05.
MajorInterval	Double	O passo entre marcas com rótulo. 0 (padrão) escolhe um passo.
MinorCount	Integer	As marcas menores entre duas marcas principais. Padrão 4.
TickPlacement	TsgcGaugeTickPlacement	gtpInside (padrão), gtpOutside ou gtpCross.
TickColor	TsgcUIColor	A cor das marcas. 0 usa o tema.
ShowLabels	Boolean	Desenha os rótulos das marcas principais. Padrão True.
LabelFormat	string	A máscara FormatFloat dos rótulos. Vazia usa as casas decimais do passo.
LabelFont	TsgcUITextStyle	A fonte dos rótulos.
Ranges	TsgcGaugeRanges	As faixas de cor.
Visible	Boolean	Desenha a escala. Padrão True.

MajorStep retorna o passo em uso e ValueToRatio(aValue) a posição de um valor de 0 a 1.

TsgcGaugeRange

Propriedade	Tipo	Descrição
StartValue, EndValue	Double	Os valores que a faixa cobre.

Color	TsgcUIColor	A cor da banda.
Thickness	Double	A largura da banda. 0 usa a espessura da escala.
Caption	string	Um nome para a faixa. Não é desenhado.

Exemplo

```
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;
```

Veja também

- Escalas
- Faixas
- TsgcCustomGauge

TsgcGaugePointer, TsgcGaugeThreshold e TsgcGaugeAnimation

Unit: sgcGauges_Model
TsgcGaugePointer é um item da coleção Pointers (TsgcGaugePointers), TsgcGaugeThreshold um item da coleção Thresholds (TsgcGaugeThresholds), e TsgcGaugeAnimation a propriedade Animation de um medidor.

TsgcGaugePointer

Propriedade	Tipo	Descrição
Kind	TsgcGaugePointerKind	gpkNeedle (padrão), gpkMarker, gpkBar ou gpkRange (uma faixa de Value até EndValue).
ScaleIndex	Integer	A escala que o ponteiro usa. Padrão 0.
Value	Double	O valor. Thread safe; escrevê-lo anima.
EndValue	Double	O fim de um ponteiro gpkRange. Thread safe.
Origin	Double	Onde começa uma barra. Não definido (NaN) significa o Minimum da escala. OriginValue retorna o valor em uso.
Color	TsgcUIColor	A cor. 0 usa o tema.
Width, Length, Offset	Double	O tamanho e a posição. Padrões 0.04, 0.9 e 0.
NeedleStyle	TsgcGaugeNeedleStyle	gndTapered (padrão), gndLine ou gndArrow.
MarkerShape	TsgcGaugeMarkerShape	gmsTriangle (padrão), gmsCircle, gmsDiamond ou gmsLine.
CapVisible	Boolean	Desenha a tampa sobre o pivô de uma agulha. Padrão True.
Visible	Boolean	Desenha o ponteiro. Padrão True.

TsgcGaugeThreshold

Propriedade	Tipo	Descrição
Value	Double	O valor que é observado.
PointerIndex	Integer	O ponteiro que é observado. Padrão 0.
Direction	TsgcGaugeThresholdDirection	gtdRising (padrão), gtdFalling ou gtdBoth.
Hysteresis	Double	Quanto o valor deve voltar antes que o limiar possa disparar novamente. Padrão 0.

Color	TsgcUIColor	A cor da marca do limiar na escala. 0 (padrão) não desenha marca.
ValueColor	TsgcUIColor	A cor do texto do valor enquanto o limiar é cruzado. Aplica-se aos limiares do ponteiro 0.
Enabled	Boolean	Padrão True.
Active	Boolean	Pública, somente leitura. True enquanto o valor está além do limiar.

TsgcGaugeAnimation

Propriedade	Tipo	Descrição
Enabled	Boolean	Anima os movimentos do ponteiro. Padrão True.
Duration	Integer	A duração de um movimento em milissegundos. Padrão 300.
Easing	TsgcGaugeEasing	geLinear, geQuadOut, geCubicOut (padrão), geCubicInOut ou geBackOut.

Exemplo

```
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;
```

Veja também

- Ponteiros

- Limiares
- Animação

Escalas

Uma escala mapeia valores para uma posição no medidor: um arco em um medidor radial, uma linha em um medidor linear. A coleção Scales de um medidor sempre tem pelo menos uma escala, de 0 a 100. Um ponteiro mostra seu valor na escala indicada por seu ScaleIndex.

Propriedades de TsgcGaugeScale

Propriedade	Descrição
Minimum, Maximum	A faixa da escala, 0 e 100 por padrão. Uma mudança da faixa limita novamente os valores dos ponteiros na escala.
Mapping	gsmLinear distribui os valores uniformemente. gsmDecibel os distribui em uma curva de decibéis, como em um VU meter.
Reversed	A escala vai de Maximum para Minimum.
Radius, Offset, Thickness	A posição e a largura da banda da escala, como frações do tamanho do medidor. Radius é 1 por padrão, a borda externa do arco, e Thickness é 0.05. Uma segunda escala com Radius igual a 0.7 fica dentro da primeira.
MajorInterval, MinorCount	O passo entre duas marcas com rótulo e o número de marcas menores entre elas (4 por padrão). Um MajorInterval igual a 0 escolhe um passo redondo para cerca de 5 intervalos. MajorStep retorna o passo em uso.
TickPlacement, TickColor	Marcas dentro da banda (gtpInside), fora dela (gtpOutside) ou atravessando-a (gtpCross), e sua cor (0 = tema).
ShowLabels, LabelFormat, LabelFont	Os rótulos das marcas principais. LabelFormat é uma máscara FormatFloat; quando está vazio, os rótulos têm as casas decimais do passo principal. Rótulos que se sobreporiam são reduzidos.
Ranges	As faixas de cor da escala. Veja Faixas.
Visible	Desenha a escala, True por padrão.

ValueToRatio retorna a posição de um valor na escala, de 0 no início a 1 no fim, com o mapeamento e Reversed aplicados.

Várias escalas

Na edição Registered, um medidor desenha todas as escalas da coleção, por exemplo km/h por fora e mph por dentro. Na Community apenas a primeira escala é desenhada. As outras escalas permanecem na coleção, de modo que um formulário feito com a edição Registered ainda carrega.

Exemplo

```
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;
```

Veja também

- Faixas
- Ponteiros
- Edições

Faixas

Uma faixa colore uma parte de uma escala, como uma banda normal verde e uma banda crítica vermelha. Cada escala tem sua própria coleção Ranges de itens TsgcGaugeRange.

Propriedades de TsgcGaugeRange

Propriedade	Descrição
StartValue, EndValue	Os valores em que a banda começa e termina, em unidades da escala. Podem ser dados em qualquer ordem.
Color	A cor da banda. 0 assume uma cor da paleta do tema, pelo índice da faixa.
Thickness	A largura da banda como fração do tamanho do medidor, desenhada a partir da borda externa da escala. 0 preenche toda a banda da escala.
Caption	Um nome para a faixa, guardado com ela. O medidor não o desenha.

Faixas e ponteiros

Um ponteiro de barra sem cor própria (Color 0) assume a cor da faixa em que seu valor está, de modo que a barra de um medidor linear fica vermelha quando o valor entra na banda vermelha. Um limiar também pode colorir o texto do valor, veja Limiares.

Exemplo

```
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;
```

Veja também

- Escalas
- Limiares
- Temas

Ponteiros

Um ponteiro mostra um valor em uma escala. A coleção Pointers de um medidor sempre tem pelo menos um ponteiro, e Value é um atalho para o valor do primeiro.

Tipos de ponteiro

Tipo	Descrição
gpkNeedle	Uma agulha a partir do centro de um medidor radial, afilada, uma linha ou uma seta (NeedleStyle), com uma tampa sobre o pivô (CapVisible). O tipo padrão.
gpkMarker	Uma pequena forma na escala: um triângulo, um círculo, um losango ou uma linha (MarkerShape). Útil para um setpoint ou um alvo.
gpkBar	Uma banda que preenche a escala de Origin até o valor.
gpkRange	Uma banda de Value até EndValue, por exemplo uma janela de tolerância.

Propriedades de TsgcGaugePointer

Propriedade	Descrição
Kind, ScaleIndex	O tipo de ponteiro e a escala em que mostra seu valor.
Value, EndValue	O valor e o fim da banda de um ponteiro de faixa. Ambos são limitados à escala e podem ser escritos a partir de qualquer thread.
Origin	Onde começa uma barra, limitada à escala. Quando não é definida, a barra começa no Minimum da escala. Um Origin igual a 0 em uma escala de -50 a 50 desenha uma barra que cresce para a esquerda ou para a direita a partir do meio.
Color	A cor do ponteiro. Com 0, uma agulha ou um marcador assume uma cor da paleta do tema, uma barra a cor da faixa em que seu valor está (ou a cor de destaque) e um ponteiro de faixa uma cor de destaque translúcida.
Width, Length, Offset	O tamanho e a posição do ponteiro, como frações do tamanho do medidor (Width 0.04 e Length 0.9 por padrão).
NeedleStyle, MarkerShape, CapVisible, Visible	A forma de uma agulha (gndTapered, gndLine, gndArrow) e de um marcador, a tampa de uma agulha e se o ponteiro é desenhado.

Vários ponteiros

Na edição Registered, todo ponteiro da coleção é desenhado, animado e avaliado quanto a limiares, por exemplo uma agulha para o valor atual e um marcador para o setpoint. Na Community apenas o primeiro ponteiro está ativo; os outros permanecem na coleção.

Exemplo

```
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;
```

Veja também

- Escalas
- Animação
- Limiares

Limiares

Um limiar observa o valor de um ponteiro e dispara OnThreshold quando o valor o cruza. Use-o para alarmes, avisos ou uma mudança de cor do texto do valor.

Como funciona um limiar

- Um limiar é avaliado quando um valor é escrito, sobre o novo valor de destino, nunca sobre os valores desenhados durante uma animação.
- gtdRising e gtdBoth ficam ativos em Value ou acima e deixam de estar ativos abaixo de Value menos Hysteresis. gtdFalling fica ativo em Value ou abaixo e deixa de estar ativo acima de Value mais Hysteresis. A histerese impede que um valor que oscila em torno do limite dispare repetidamente.
- Com gtdRising e gtdFalling, OnThreshold dispara quando o limiar se torna ativo. Com gtdBoth dispara nas duas travessias, e aRising indica a direção. Para um alarme que também precisa de um evento de término, use gtdBoth com uma histerese.
- Uma mudança de configuração, como um novo Value do limiar, atualiza Active sem evento.

Propriedades de TsgcGaugeThreshold

Propriedade	Descrição
Value, Hysteresis, Direction	O limite, a banda morta e a direção (gtdRising por padrão).
PointerIndex	O ponteiro que o limiar observa, 0 por padrão.
Color	Quando não é 0, uma marca desta cor é desenhada na escala em Value.
ValueColor	Quando não é 0, o texto do valor assume esta cor enquanto o limiar está ativo (limiares do primeiro ponteiro).
Enabled, Active	Desliga o limiar e informa se ele está ativo agora (somente leitura).

OnThreshold

`procedure(Sender: TObject; aThreshold: TsgcGaugeThreshold; aValue: Double; aRising: Boolean)`. Os controles VCL e FireMonkey a chamam na thread principal, na ordem das travessias, seja qual for a thread que escreveu o valor. Um modelo sem host a chama na thread que escreveu o valor.

Exemplo

```
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;
```

Veja também

- Faixas
- Ponteiros
- Início rápido VCL

Animação

Quando um valor muda, o ponteiro se move do valor que mostra para o novo em um curto tempo, com uma curva de suavização. O valor em si, os limiares e os eventos mudam de uma vez; só o desenho se move.

Propriedades de TsgcGaugeAnimation

Propriedade	Descrição
Enabled	Anima as mudanças de valor, True por padrão.
Duration	O tempo de um movimento em milissegundos, 300 por padrão. 0 desliga a animação.
Easing	A curva do movimento: geLinear, geQuadOut, geCubicOut (o padrão), geCubicInOut ou geBackOut, que ultrapassa um pouco no final.

Quando um valor não anima

- Em tempo de projeto, com Enabled definido como False ou com Duration igual a 0.
- Quando o valor é escrito com `Gauge.SetValue(aPointer, aValue, False)`.
- Quando o valor é carregado de um arquivo de formulário, enquanto a coleção Pointers está sendo atualizada.
- Em uma renderização sem janela, que sempre desenha o valor de destino.

Um novo valor que chega durante um movimento parte do valor desenhado naquele instante, de modo que o ponteiro nunca dá saltos.

Quadros

Os controles VCL e FireMonkey desenham os quadros de um movimento apenas na camada dinâmica, e no máximo MaxFPS vezes por segundo (60 por padrão). Quando nada se move, um medidor não desenha nenhum quadro. A hora do medidor vem da função Clock do modelo; DisplayValue retorna o valor desenhado em uma determinada hora e IsAnimating informa se um ponteiro ainda está se movendo.

Exemplo

```
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;
```

Veja também

- Ponteiros
- Arquitetura
- Saída sem janela

Temas

Um medidor toma suas cores do tema do sgcUICore definido em **Theme**: utLight (o padrão) ou utDark. Toda propriedade de cor de um medidor é 0 por padrão, o que significa a cor do tema, de modo que um medidor sem cores próprias segue totalmente o tema.

Cores

Propriedade	Com 0
BackgroundColor	O fundo do tema.
FaceColor	A cor do mostrador do tema, atrás da escala de um mostrador.
Scales[i].TickColor	A cor de texto atenuada do tema.
Ranges[i].Color, Pointers[i].Color	Cores da paleta do tema, veja Faixas e Ponteiros.
TitleFont.Color, ValueFont.Color, LabelFont.Color	A cor de texto do tema.

As cores são escritas como $AARRGGBB, por exemplo $FFE34948 para um vermelho opaco.

Texto

- TitleFont, ValueFont e o LabelFont de cada escala são valores TsgcUITextStyle com Size, Bold e Color. Os tamanhos são relativos ao medidor: o texto cresce e diminui com o controle. O texto do valor é desenhado com o dobro do tamanho de ValueFont.
- ValueFormat é uma máscara FormatFloat para o texto do valor. Quando está vazia, o valor tem as casas decimais da escala. Units é desenhado ao lado do valor.
- Os números usam o FormatSettings do modelo, com '.' como separador decimal por padrão, de modo que um medidor tem o mesmo aspecto em qualquer localidade até você alterá-lo.
- OnFormatValue permite substituir o texto do valor de qualquer ponteiro.

Exemplo

```
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;
```

Veja também

- Faixas
- Início rápido FMX
- Saída sem janela

Saída sem janela

Todo medidor é renderizado sem janela. As classes de modelo desenham em um canvas do sgcUICore e salvam arquivos PNG e SVG, de modo que um servidor, um relatório ou um programa de console no Windows ou no Linux64 pode produzir as mesmas imagens que os controles.

Métodos de um modelo

Método	Descrição
SaveToPNG(aStream or aFileName, aWidth, aHeight, aScale = 1)	Uma imagem PNG de aWidth x aHeight pixels lógicos, multiplicada por aScale. Uma escala de 2 dá uma imagem do dobro do tamanho para uma tela de DPI alto ou uma impressão.
SaveToSVG(aStream or aFileName, aWidth, aHeight, aScale = 1)	Um documento SVG do mesmo desenho. Os números no SVG nunca dependem da localidade.
Render(aCanvas, aWidth, aHeight, aScale = 1, aNowMs = -1)	Desenha as duas camadas em um TsgcUICanvas. Com aNowMs abaixo de 0, os ponteiros são desenhados em seus valores de destino.
RenderStatic, RenderDynamic	Desenham apenas a camada estática ou apenas a camada dinâmica.

Controles

Os controles VCL e FireMonkey têm SaveToPNG(aStream or aFileName, aScale = 1) e SaveToSVG com os mesmos argumentos. Eles salvam o medidor no tamanho de cliente do controle, de modo que uma captura de um painel não precisa de captura de janela.

A mesma saída em qualquer lugar

A saída sem janela é determinística: o mesmo medidor gera arquivos PNG e SVG idênticos byte a byte no Win32, no Win64 e no Linux64. As units do modelo não usam VCL nem FireMonkey e, no Linux64, compilam a partir do Delphi 10.2.

Exemplo

```
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;
```

Veja também

- Início rápido sem janela
- Plataformas
- Temas

Edições

sgcGauges é compilado a partir do mesmo código-fonte para todas as edições.

Edição	Inclui
Registered	Tudo: os dez tipos de medidor para VCL e FireMonkey, várias escalas e ponteiros por medidor, os displays de 14 segmentos e de matriz de pontos, e a ponte sgcGaugesWS para o sgcWebSockets.
Trial	Os mesmos recursos da Registered. Um build compilado com SGC_TRIAL mostra uma mensagem de que é uma versão de avaliação quando um componente é criado depois de passada a janela de lançamento indicada no build. Nada mais muda.
Community	Seis tipos de medidor para VCL e FireMonkey sem os recursos avançados listados abaixo, e sem aviso de avaliação.

O que a Community inclui

O medidor radial (todos os estilos), o medidor linear (barra, termômetro e bullet), o display digital (7 segmentos), o LED, a barra de LEDs e o anel de progresso, com faixas, limiares, animação, os temas claro e escuro, os controles VCL e FireMonkey e a saída PNG e SVG sem janela. O alvo bullet do medidor linear e o pico da barra de LEDs são propriedades, não ponteiros extras, de modo que funcionam na Community.

Os recursos avançados

A define SGC_GAUGES_ADVANCED em sgcVer.inc os ativa. Ela está desativada na edição Community.

- **Várias escalas e ponteiros**: as escalas e os ponteiros além do primeiro são mantidos em suas coleções, de modo que um formulário feito com a edição Registered ainda carrega, mas a Community não os desenha, não os anima nem avalia limiares para eles.
- **Quatro tipos de medidor**: o VU meter, o relógio, a bússola e o anunciador, suas units (sgcGauges_VUMeter, sgcGauges_Clock, sgcGauges_Compass, sgcGauges_Annunciator), seus oito controles e seu registro.
- **Displays digitais**: o desenho de 14 segmentos e de matriz de pontos. gdt14Segment e gdtDotMatrix permanecem na enumeração e são desenhados como 7 segmentos.
- **A ponte**: os pacotes sgcGaugesWS não são compilados nem instalados na Community.

Código para todas as edições

O código que usa um tipo avançado pode testar a define, de modo que a mesma unit compila nas duas edições.

```
{$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;
```

Veja também

- Tipos de medidor
- Instalação
- Solução de problemas

Visão geral da ponte

sgcGaugesWS é a ponte entre sgcGauges e sgcWebSockets. Suas fontes leem as mensagens JSON que um cliente sgcWebSockets recebe e escrevem os valores nos medidores, sem código nos eventos de mensagem. A ponte faz parte da edição Registered e precisa do sgcWebSockets; tem seus próprios pacotes, sgcGaugesWS e dclsgcGaugesWS, e seus componentes estão na página **SGC Gauges** da paleta.

Componente	Unit	Cliente
TsgcGaugeFeedSource_MQTT	sgcGaugesWS_Source_MQTT	TsgcWSPClient_MQTT
TsgcGaugeFeedSource_WebSocket	sgcGaugesWS_Source_WebSocket	TsgcWebSocketClient
TsgcGaugeFeedSource_SSE	sgcGaugesWS_Source_SSE	TsgcHTTP1Client

Como funciona uma fonte

Enquanto uma fonte está Active e seu Client está atribuído, em tempo de execução, ela instala seu próprio manipulador de mensagens no cliente e mantém o manipulador que existia antes, de modo que o manipulador da aplicação ainda dispara depois da fonte. Definir Active como False, limpar Client ou liberar a fonte restaura o manipulador anterior. A fonte não conecta, não assina nem reconecta: a aplicação faz isso com o cliente.

Vínculos

Cada item de Bindings (TsgcGaugeFeedBinding, unit sgcGaugesWS_Source) escreve um ponteiro de um medidor. Uma mensagem é analisada uma única vez como JSON e lida por todos os vínculos ativos.

Propriedade	Tipo	Descrição
Gauge	TComponent	Um controle de medidor VCL ou FireMonkey. Outros componentes são rejeitados.
GaugeModel	TsgcCustomGauge	Pública. Um modelo sem janela; tem prioridade sobre Gauge. Limpe-a antes de o modelo ser liberado.
PointerIndex	Integer	O ponteiro escrito. Padrão 0.
ValuePath	string	O caminho JSON do valor, como price ou data.items[0].v. Vazio lê a mensagem inteira.
FilterPath	string	O caminho de um campo que seleciona as mensagens. O vínculo só lê uma mensagem quando o caminho existe.
FilterValue	string	Quando não vazio, o texto de FilterPath deve ser igual a ele, como ticker.
Topic	string	Somente MQTT: um filtro de tópico com + e #. Vazio lê todas as publicações.
EventName	string	Somente SSE: o tipo de evento. Vazio lê todos os eventos.
Active	Boolean	False faz o vínculo ignorar todas as mensagens. Padrão True.

Um número JSON, uma string numérica com '.' como separador decimal e um booleano (true é 1, false é 0) são aceitos. Uma mensagem sem o caminho de um vínculo é ignorada por esse vínculo, de modo que atualizações parciais funcionam. JSON inválido, um valor de outro tipo ou um índice de ponteiro inválido vão para OnError.

Propriedades e eventos da fonte

Membro	Descrição
Client	O cliente sgcWebSockets que a fonte lê.
Active	False faz a fonte ignorar todas as mensagens. Padrão True.
Bindings	Os vínculos.
OnValue(Sender, aBinding, aValue)	Um vínculo escreveu um valor.
OnError(Sender, aError)	Não foi possível aplicar uma mensagem.

Threads

Os valores são escritos com o SetValue thread safe do medidor, na thread em que a mensagem chega, e os controles de medidor se repintam e chamam OnThreshold na thread principal. As escritas de uma mensagem em um medidor viram uma única mudança. OnValue e OnError rodam na thread de chegada sob o lock da fonte: use queue (TThread.Queue) em vez de synchronize a partir deles antes de tocar em um controle. Altere Bindings a partir da thread principal.

Compilação

Mantenha as pastas de código-fonte do sgcGauges e do sgcWebSockets separadas no caminho de busca. Cada pasta tem seu próprio sgcVer.inc, e uma unit lê o da sua própria pasta.

Exemplo

Cada fonte tem um método público que aplica uma mensagem como se o cliente a tivesse recebido: ProcessPublish (MQTT), ProcessMessage (WebSocket) e ProcessChunk (SSE). Eles retornam a contagem de valores escritos. Isto alimenta um modelo sem janela, sem 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;
```

Veja também

- Fonte MQTT
- Fonte WebSocket
- Fonte de eventos enviados pelo servidor
- TsgcCustomGauge

Fonte MQTT

Classe: TsgcGaugeFeedSource_MQTT (unit sgcGaugesWS_Source_MQTT)
Alimenta medidores com as publicações que um TsgcWSPClient_MQTT recebe. Client é o cliente MQTT. A fonte instala seu manipulador OnMQTTPublish e mantém o anterior, e o gancho continua funcionando entre reconexões. O Topic de cada vínculo é um filtro de tópico MQTT: + corresponde a um nível e # a este nível e a todos os inferiores. Um vínculo lê apenas as publicações cujo tópico corresponde a ele, e um Topic vazio lê todas as publicações. A fonte não assina, então a aplicação assina no cliente, com um filtro que cubra os tópicos dos vínculos. Todo payload é lido como JSON.
ProcessPublish(aTopic, aText) aplica uma publicação sem cliente e retorna a contagem de valores escritos. A função sgcGaugeMQTTTopicMatch(aFilter, aTopic) da mesma unit testa um tópico contra um filtro.

Exemplo

Dois medidores radiais alimentados pelos tópicos sensors/<id>/temperature e sensors/<id>/humidity, com 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;
```

Veja também

- Visão geral da ponte
- Fonte WebSocket
- Fonte de eventos enviados pelo servidor

Fonte WebSocket

Classe: TsgcGaugeFeedSource_WebSocket (unit sgcGaugesWS_Source_WebSocket)
Alimenta medidores com as mensagens de texto que um TsgcWebSocketClient recebe. Client é o cliente WebSocket. A fonte instala seu manipulador OnMessage e mantém o anterior, e o gancho continua funcionando entre reconexões, como as do WatchDog. Toda mensagem de texto é lida como JSON com ValuePath, FilterPath e FilterValue dos vínculos, na thread a partir da qual o cliente notifica, de modo que qualquer modo NotifyEvents do cliente é seguro.
FilterPath e FilterValue selecionam as mensagens de um fluxo que carrega vários tipos, como um canal de ticker e outro de negociações em uma só conexão. ProcessMessage(aText) aplica uma mensagem sem cliente e retorna a contagem de valores escritos.

Exemplo

Um display de preço e uma barra de variação alimentados pelas mensagens ticker de um fluxo:

```
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;
```

Veja também

- Visão geral da ponte
- Fonte MQTT
- Fonte de eventos enviados pelo servidor

Fonte de eventos enviados pelo servidor

Classe: TsgcGaugeFeedSource_SSE (unit sgcGaugesWS_Source_SSE)
Alimenta medidores com os eventos enviados pelo servidor de um TsgcHTTP1Client. Client é o cliente HTTP. A fonte instala seu manipulador OnSSEMessage e mantém o anterior. A aplicação inicia o fluxo com Client.GetSSE. A fonte divide os chunks do fluxo em eventos: as linhas data: são unidas com um line feed, uma linha event: define o tipo do evento e uma linha vazia encerra o evento. Finais de linha LF, CRLF e CR são aceitos, e um evento pode ser dividido entre chunks. O EventName de cada vínculo seleciona o tipo de evento que ele lê. Um evento sem linha event: tem o tipo message, e um EventName vazio lê todos os eventos. Os dados de cada evento são lidos como JSON.

Método	Descrição
ResetStream	Descarta um evento que um fluxo deixou meio lido. Chame-o antes de um novo fluxo após uma reconexão.
ProcessChunk(aText)	Aplica um chunk de um fluxo sem cliente e retorna a contagem de valores escritos.

Exemplo

Um anel de progresso e uma barra de LEDs alimentados pelos eventos cpu e queue de um fluxo 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;
```

Veja também

- Visão geral da ponte
- Fonte MQTT
- Fonte WebSocket

Visão geral do .NET

sgcGauges .NET é uma versão C# gerenciada da biblioteca sgcGauges do Delphi, não um wrapper em torno dela. As classes ficam no namespace **esegece.sgcGauges** e mantêm os nomes do Delphi: **TsgcCustomGauge**, os dez modelos de medidor como **TsgcRadialGauge** e **TsgcLEDBarGauge**, **TsgcGaugeScale**, **TsgcGaugeRange**, **TsgcGaugePointer**, **TsgcGaugeThreshold**, **TsgcGaugeAnimation** e as enumerações com seus prefixos de medidor (**grsHalf**, **gtdBoth**, **glsBullet**). Os controles seguem o padrão de nomes do Delphi em todos os hosts: **TsgcVCLRadialGauge** é **TsgcWinFormsRadialGauge**, **TsgcWpfRadialGauge** e **TsgcAvaloniaRadialGauge**. Os medidores são desenhados com a versão .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos demais produtos da suíte visual da eSeGeCe.
Como no Delphi, cada controle possui um modelo independente de framework, exposto como **Gauge**. O modelo guarda as escalas, faixas, ponteiros, limiares e a animação e se desenha em qualquer canvas do sgcUICore, portanto o mesmo código de modelo funciona em um formulário WinForms, em uma janela WPF ou Avalonia e em um programa de console sem interface, veja Renderização sem janela no .NET.

Pacotes

A versão se divide em seis pacotes NuGet, para que um projeto traga apenas o que usa.

PackageId	Contém	Frameworks de destino
esegece.sgcGauges	Os dez modelos de medidor, as classes de escala, faixa, ponteiro, limiar e animação, o painter e a saída SVG (**SaveToSVG**). Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGauges.Rendering	A saída PNG sem janela: os métodos de extensão **SaveToPNG** e **RenderToRaster** de **sgcGauges_Raster**, desenhados no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, Linux e macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcGauges.WinForms	Os dez controles **TsgcWinForms<Type>Gauge** sobre **TsgcWinFormsCustomGauge**.	net472, net8.0-windows, net9.0-windows
esegece.sgcGauges.Wpf	Os dez controles **TsgcWpf<Type>Gauge** sobre **TsgcWpfCustomGauge**.	net472, net8.0-windows, net9.0-windows
esegece.sgcGauges.Avalonia	Os dez controles **TsgcAvalonia<Type>Gauge** sobre **TsgcAvaloniaCustomGauge**, no Windows, Linux e macOS com Avalonia 11.	net8.0, net9.0
esegece.sgcGauges.sgcWebSockets	A ponte para o sgcWebSockets: **TsgcGaugeFeedSource_WebSocket**, **TsgcGaugeFeedSource_MQTT** e **TsgcGaugeFeedSource_SSE**. Somente na edição Registered.	netstandard2.0, net8.0, net9.0

Uma linha `PackageReference` traz tudo o que um host precisa: os pacotes WinForms e Avalonia dependem de esegece.sgcGauges.Rendering, que depende de esegece.sgcGauges; o pacote WPF depende de esegece.sgcGauges e desenha por meio de esegece.sgcUICore.Wpf. Veja Instalação pelo NuGet.

Edições

Edição	Ids dos pacotes	Inclui
Registered	esegece.sgcGauges, .Rendering, .WinForms, .Wpf, .Avalonia e .sgcWebSockets	Tudo: os dez tipos de medidor nos três hosts e sem janela, várias escalas e ponteiros por medidor, os 14 displays de segmentos e matriz de pontos, e a ponte sgcWebSockets.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcGauges.WinForms.Community. Não há pacote de ponte Community.	Os medidores radial, linear, digital, LED, barra de LED e anel de progresso. O VU meter, o relógio, a bússola e o anunciador não estão nos assemblies Community; somente a primeira escala e o primeiro ponteiro de um medidor são desenhados, animados e verificados quanto a limiares; **gdt14Segment** e **gdtDotMatrix** são desenhados como 7 segmentos.

É a mesma divisão das edições do Delphi, veja Edições.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	VCL (controles TsgcVCL)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	VCL e FMX	Windows, .NET Framework 4.7.2 e .NET 8 e 9. Os controles podem ser declarados em XAML.
Avalonia	FMX (controles TsgcFMX)	Windows, Linux e macOS, .NET 8 e 9.
Sem janela	Os modelos de medidor	Qualquer plataforma dos frameworks de destino: SVG com esegece.sgcGauges, PNG com esegece.sgcGauges.Rendering.

Todo controle publica as propriedades de seu equivalente Delphi com os mesmos nomes, além de **Gauge**, **Theme**, **OnThreshold**, **MaxFPS**, **SaveToPNG** e **SaveToSVG** na classe base. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia

- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Vinculação de dados no .NET
- Renderização sem janela no .NET
- Ponte sgcWebSockets no .NET
- Demos do .NET
- Plataformas

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, os pacotes do núcleo e da renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcGauges.WinForms
dotnet add package esegece.sgcGauges.Wpf
dotnet add package esegece.sgcGauges.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcGauges.WinForms.Community
dotnet add package esegece.sgcGauges.Wpf.Community
dotnet add package esegece.sgcGauges.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcGauges.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie os pacotes Registered ou os pacotes Community em um projeto, nunca os dois: eles contêm os mesmos tipos no mesmo namespace.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace em XAML com `xmlns:sgc="clr-namespace:esegece.sgcGauges;assembly=esegece.sgcGauges.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como em qualquer aplicativo desktop Avalonia. Mapeie o namespace em XAML com `xmlns:sgc="using:esegece.sgcGauges"`.
- Sem janela: esegece.sgcGauges sozinho basta para os modelos e o SVG. Adicione esegece.sgcGauges.Rendering para PNG, que traz o SkiaSharp e seus assets nativos para Windows e Linux.
- Dados ao vivo do sgcWebSockets: adicione **esegece.sgcGauges.sgcWebSockets** (edição Registered), que depende de esegece.sgcGauges e esegece.sgcWebSockets, veja Ponte sgcWebSockets no .NET.

Em C#, todo host precisa de `using esegece.sgcGauges;`, e de `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUITextStyle** e os demais tipos do sgcUICore.

Veja também

- Visão geral do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Demos do .NET

Início rápido WinForms

Este início rápido cria em código um formulário com três medidores WinForms: um velocímetro com uma faixa vermelha e um limiar em 200 km/h, uma barra de combustível e uma lâmpada de excesso de velocidade que o limiar liga e desliga.

Passos

1. Crie um projeto Windows Forms para `net8.0-windows` e adicione o pacote **esegece.sgcGauges.WinForms** (**esegece.sgcGauges.WinForms.Community** na edição Community).
2. Adicione `using esegece.sgcGauges;` e `using esegece.sgcUICore;`.
3. Crie os controles e adicione-os ao formulário. São controles WinForms comuns: Bounds, Dock, Anchor e Enabled funcionam como de costume.
4. Defina primeiro o intervalo da escala, depois as faixas, os limiares e o valor: um valor é limitado à escala quando é gravado.
5. Atribua os eventos. Eles têm os nomes do Delphi e são atribuídos com `=`.

Exemplo

Notas

- **Value** é **Pointers[0].Value**. Qualquer thread pode gravá-lo: o controle se redesenha na thread da interface e dispara **OnThreshold** nela.
- Com **Animation.Enabled** (o padrão), a agulha se move para o novo valor durante **Animation.Duration** milissegundos; o valor, os limiares e os eventos mudam de uma vez. Somente o ponteiro em movimento é redesenhado, no máximo **MaxFPS** quadros por segundo.
- **Theme** é por controle; uma propriedade de cor deixada em 0 usa a cor do tema.
- **SaveToPNG** e **SaveToSVG** gravam o que o controle mostra em seu tamanho, com uma escala opcional.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Vinculação de dados no .NET
- Início rápido VCL

Início rápido WPF

Este início rápido declara em XAML um velocímetro, um display digital de rpm e uma lâmpada de excesso de velocidade, e vincula os dois valores a um view model.

Passos

1. Crie um projeto WPF para `net8.0-windows` e adicione o pacote **esegece.sgcGauges.Wpf** (**esegece.sgcGauges.Wpf.Community** na edição Community).
2. Mapeie o namespace na janela: `xmlns:sgc="clr-namespace:esegece.sgcGauges;assembly=esegece.sgcGauges.Wpf"`.
3. Declare os controles. Propriedades simples como **Title**, **Units**, **ValueFormat** e **Digits** são definidas como atributos; **Value** é uma propriedade de dependência e aceita um vínculo.
4. Preencha as coleções (escalas, faixas, limiares) e atribua os eventos no 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 controle tem uma propriedade de dependência: **ValueProperty** nos medidores radial, linear, digital, LED, barra de LED, anel de progresso e VU meter e **HeadingProperty** na bússola. O relógio e o anunciador não têm nenhuma. As demais propriedades são propriedades CLR. Veja Vinculação de dados no .NET.
- **OnThreshold** é executado na thread da interface, também quando o valor vinculado é definido a partir de outra thread.
- **Theme** é por controle.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Vinculação de dados no .NET
- Início rápido FMX

Início rápido Avalonia

Este início rápido cria em código uma janela Avalonia com um mostrador de meia volta para uma pressão, um anel de progresso e uma barra de LED com faixas verde, âmbar e vermelha, tudo no tema escuro. O mesmo código funciona no Windows, Linux e macOS.

Passos

1. Crie um projeto `net8.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** (Avalonia 11.3) e adicione **esegece.sgcGauges.Avalonia** (**esegece.sgcGauges.Avalonia.Community** na edição Community).
2. Adicione `using esegece.sgcGauges;` e `using esegece.sgcUICore;`.
3. Crie os controles, defina a escala e o valor e adicione-os a um painel da janela.

Exemplo

Notas

- Os controles também podem ser declarados em XAML com `xmlns:sgc="using:esegece.sgcGauges"`, por exemplo `<sgc:TsgcAvaloniaRadialGauge Title="Pressure" Value="{Binding Pressure}" />`.
- **ValueProperty** (todos os controles), **ActiveProperty** (LED) e **HeadingProperty** (bússola) são propriedades estilizadas que se vinculam nos dois sentidos por padrão, veja Vinculação de dados no .NET.
- O timer de animação de um controle é interrompido quando ele sai da árvore visual; os controles não precisam de Dispose.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Vinculação de dados no .NET
- Início rápido FMX

Classes do .NET

As classes de modelo têm os nomes do Delphi e ficam no namespace **esegece.sgcGauges**. Os controles trocam o prefixo VCL e FMX pelo nome do host.

Medidores e 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

Os quatro tipos Registered e seus controles não estão nos assemblies Community. Veja Tipos de medidor para o que cada tipo desenha.

Classes base

Classe	Pacote	Descrição
TsgcCustomGauge	esegece.sgcGauges	A base do modelo: **Scales**, **Pointers**, **Thresholds**, **Animation**, **Theme**, **Title**, **Units**, **ValueFormat**, **FormatSettings**, **Value**, **SetValue**, **ValueToText**, **BeginUpdate** e **EndUpdate**, **Render**, **RenderS-**

		tatic, **RenderDynamic**, **SaveToSVG** e os eventos **OnChange**, **OnThreshold** e **OnFormatValue**.
TsgcWinFormsCustomGauge	esegece.sgcGauges.WinForms	A base dos controles WinForms, sobre **TsgcUIWinFormsControl**.
TsgcWpfCustomGauge	esegece.sgcGauges.Wpf	A base dos controles WPF, sobre **TsgcUIWpfControl**.
TsgcAvaloniaCustomGauge	esegece.sgcGauges.Avalonia	A base dos controles Avalonia, sobre **TsgcUIAvaloniaControl**, com **ValueProperty**, **LogicalWidth()** e **LogicalHeight()**.
IsgcGaugeHost	esegece.sgcGauges	Implementada por todos os controles: **GetGauge()** retorna o modelo. A ponte controla qualquer objeto que a implemente.
sgcGauges_Raster	esegece.sgcGauges.Rendering	Os métodos de extensão **SaveToPNG** e **RenderToRaster** de **TsgcCustomGauge**.

Partes do modelo

Classe	Descrição
TsgcGaugeScale, TsgcGaugeScales	Uma escala: **Minimum**, **Maximum**, **Mapping**, **MajorInterval**, **MinorCount**, **TickPlacement**, **LabelFormat**, **Ranges**.
TsgcGaugeRange, TsgcGaugeRanges	Uma faixa colorida de uma escala: **StartValue**, **EndValue**, **Color**, **Thickness**.
TsgcGaugePointer, TsgcGaugePointers	Um ponteiro de tipo agulha, marcador, barra ou faixa: **Kind**, **ScaleIndex**, **Value**, **EndValue**, **Origin**, **Color**, **NeedleStyle**, **MarkerShape**.
TsgcGaugeThreshold, TsgcGaugeThresholds	Um limite que dispara **OnThreshold**: **Value**, **PointerIndex**, **Direction**, **Hysteresis**, **Color**, **ValueColor**, **Enabled**, **Active**.
TsgcGaugeAnimation	A animação dos ponteiros: **Enabled**, **Duration** em milissegundos e **Easing**.
TsgcGaugeAnnunciatorTile, TsgcGaugeAnnunciatorTiles	Um bloco do anunciador: **Caption**, **State**, **Acknowledged**.
TsgcGaugeFormatSettings	Uma struct com o **DecimalSeparator** e o **ThousandSeparator** dos rótulos, '.' e ',' por padrão.

Delegates

Delegate	Assinatura
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)`, o **OnTileClick** dos controles anunciador

As classes da ponte estão listadas em Ponte sgcWebSockets no .NET.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcCustomGauge
- Tipos de medidor

Diferenças do .NET em relação ao Delphi

A versão mantém os nomes, os padrões e o comportamento do Delphi, portanto os tópicos Delphi desta ajuda também valem para o .NET. Estas são as diferenças que vêm da linguagem e dos frameworks.

Linguagem

- **Eventos** são propriedades delegate com os nomes do Delphi (**OnThreshold**, **OnFormatValue**, **OnChange**, **OnTileClick**). Atribua-os com `=`: um manipulador por evento, como no Delphi. O parâmetro `var` de **OnFormatValue** é um parâmetro `ref`. Nos eventos de um controle, Sender é o controle.
- **Funções sem parâmetros** continuam sendo métodos: **ActiveScaleCount()**, **ActivePointerCount()**, **NowMs()**, **GetGauge()**, **TsgcGaugeAnnunciatorTile.Flashing()**.
- **FormatSettings** é a struct **TsgcGaugeFormatSettings**: leia, altere e atribua de volta.
- **Cores** são valores `uint` na ordem ARGB (`0xFFE34948`). 0 significa a cor do tema, como no Delphi.
- **Coleções** derivam de **TsgcGaugeCollection<T>**: um indexador (`Scales[0]`), **Add**, **Delete**, **Clear**, **Count**, **BeginUpdate** e **EndUpdate**, e `foreach`. **Assign(object)** copia uma coleção ou um item.
- As funções livres das units do Delphi são métodos estáticos: **sgcGauges_Types** (matemática e easing), **sgcGauges_Render** (geometria) e **sgcGauges_Raster** (PNG).
- Não existe **Free**: os modelos são classes comuns. Os controles WinForms e WPF seguem seu framework; um controle Avalonia interrompe seu timer ao sair da árvore visual.

Exemplo

Saída

- **SaveToSVG** é um método do modelo em esegece.sgcGauges. **SaveToPNG** de um modelo é um método de extensão em esegece.sgcGauges.Rendering, porque o PNG precisa do raster SkiaSharp. Os controles têm os dois, em seu próprio tamanho.
- O SVG de um medidor tem os mesmos bytes que o SVG do Delphi do mesmo medidor. O PNG é desenhado pelo SkiaSharp e pode diferir do PNG do Delphi no antialiasing das bordas.

Threads e tempo

- **SetValue** e **Value** são thread safe, como no Delphi. Os controles se redesenham e disparam **OnThreshold** na thread da interface por meio do **SynchronizationContext** capturado na criação, no lugar de **TThread.Queue**. Em um modelo sem controle, **OnThreshold** é executado na thread que gravou o valor.
- O relógio da animação é um **Stopwatch** no lugar de **sgcGaugeNowMs**; **Clock** pode substituí-lo por qualquer **TsgcGaugeClockFunc**.

Não portado

- Os auxiliares de streaming de .dfm (as funções **Is*Stored**, as propriedades definidas ZeroValues e **GetNamePath**) não têm equivalente no .NET. As coleções são preenchidas em código.

Veja também

- Visão geral do .NET
- Classes do .NET
- Renderização sem janela no .NET
- TsgcCustomGauge
- Edições

Vinculação de dados no .NET

Um medidor mostra um valor, portanto a vinculação de dados consiste em levar um valor de um objeto do aplicativo para o controle. Cada host faz isso com seu próprio sistema de vinculação, na propriedade de valor do controle. Escalas, faixas, ponteiros e limiares são coleções e são definidos em código.

Host	Propriedades vinculáveis	Direção padrão
WinForms	**Value** (e toda propriedade pública) por meio de **Control.DataBindings**	Da fonte para o controle com **DataSourceUpdateMode.Never**; os controles não disparam evento de alteração.
WPF	Propriedades de dependência **ValueProperty** (radial, linear, digital, LED, barra de LED, anel de progresso, VU meter) e **HeadingProperty** (bússola)	Em uma direção. Com **BindingMode.TwoWay**, um valor que o modelo limita ou ajusta circularmente é gravado de volta na fonte.
Avalonia	Propriedades estilizadas **ValueProperty** (todos os controles), **ActiveProperty** (LED) e **HeadingProperty** (bússola)	Nos dois sentidos: um valor que o modelo limita ou ajusta circularmente é gravado de volta na fonte.

A fonte pode mudar em qualquer thread: o modelo é thread safe e o controle se redesenha na thread da interface. Os próprios vínculos do WinForms devem ser disparados na thread da interface, como em qualquer controle WinForms.

WinForms

WPF

Em XAML: `<sgc:TsgcWpfRadialGauge Value="{Binding Speed}" />`, veja Início rápido WPF. Em código:

Avalonia

Feeds

Para controlar medidores com mensagens de WebSocket, MQTT ou eventos enviados pelo servidor sem um view model, vincule-os a uma fonte de dados da ponte, veja Ponte sgcWebSockets no .NET.

Veja também

- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Ponte sgcWebSockets no .NET

Renderização sem janela no .NET

Todo modelo de medidor se desenha sem controle, portanto um programa de console, um serviço ou um servidor web pode gravar imagens PNG e SVG de medidores, no Windows, Linux e macOS. Esta é a forma .NET de Saída sem janela.

Métodos

Método	Pacote	Descrição
SaveToSVG(Stream or string, int aWidth, int aHeight, double aScale = 1)	esegece.sgcGauges	Grava um documento SVG do medidor. Sem dependência nativa.
SaveToPNG(Stream or string, int aWidth, int aHeight, double aScale = 1)	esegece.sgcGauges.Rendering	Método de extensão de **sgcGauges_Raster**: desenha em um raster SkiaSharp limpo de Round(aWidth * aScale) por Round(aHeight * aScale) pixels e grava um PNG.
RenderToRaster(int aWidth, int aHeight, double aScale = 1, double aNowMs = -1)	esegece.sgcGauges.Rendering	Retorna o **TsgcUICanvas_Raster** com o medidor desenhado, para desenhar mais ou ler pixels. O chamador o descarta.
Render(TsgcUICanvas aCanvas, int aWidth, int aHeight, double aScale = 1, double aNowMs = -1)	esegece.sgcGauges	Desenha em qualquer canvas do sgcUICore.

Exemplo

Notas

- Uma renderização sem janela desenha os valores de destino, nunca um quadro da animação, portanto a saída do mesmo medidor sempre tem os mesmos bytes. Um relógio com **Live** definido como false mostra seu **Time**.
- aScale multiplica os pixels e mantém o layout: 200 por 200 na escala 2 é um PNG de 400 por 400 do mesmo desenho.
- No Linux, esegece.sgcGauges.Rendering traz os assets nativos do SkiaSharp; não é necessário servidor X.

A demonstração ConsoleRender grava um arquivo PNG e um SVG de cada tipo de medidor, veja Demos do .NET.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- Demos do .NET
- Saída sem janela
- Início rápido sem janela

Ponte sgcWebSockets no .NET

O pacote **esegece.sgcGauges.sgcWebSockets** (edição Registered) é a versão .NET da ponte sgcGaugesWS. Uma fonte de dados se conecta a um cliente do sgcWebSockets .NET, lê cada mensagem como JSON e grava os valores de seus vínculos nos medidores. O comportamento é o descrito em Visão geral da ponte.

Classe	Cliente	Lê
TsgcGaugeFeedSource_WebSocket	**Client**: um **TsgcWebSocketClient**, seu evento **OnMessage**	Cada mensagem de texto. **ProcessMessage(string)** aplica uma mensagem manualmente.
TsgcGaugeFeedSource_MQTT	**Client**: um **TsgcWSProtocol_MQTT_Client**, seu evento de publicação	O conteúdo das publicações cujo tópico corresponde ao filtro **Topic** de um vínculo (curingas `+` e `#`). **ProcessPublish(topic, text)** aplica uma manualmente.
TsgcGaugeFeedSource_SSE	**Client**: um **TsgcHTTP1Client**, seu evento **OnSSEMessage**	Os dados dos eventos cujo tipo é igual ao **EventName** de um vínculo. **ProcessChunk(string)** e **ResetStream()** aplicam um fluxo manualmente.

Toda fonte tem **Active**, **Bindings**, **OnValue** e **OnError**. Um **TsgcGaugeFeedBinding** tem **Gauge** (qualquer controle, ou qualquer objeto que implemente **IsgcGaugeHost**), **GaugeModel** (um modelo sem controle), **PointerIndex**, **ValuePath**, **FilterPath** e **FilterValue**, **Topic**, **EventName** e **Active**.

Exemplo

Notas

- As fontes nunca se conectam nem assinam: o aplicativo conecta o cliente, assina os tópicos MQTT e chama **GetSSE** do cliente HTTP.
- Os eventos do cliente são eventos .NET: uma fonte adiciona seu próprio manipulador com `+=` e remove somente esse, de modo que os manipuladores do aplicativo continuam funcionando e são executados na ordem em que foram adicionados.
- Os valores são gravados na thread do cliente. Os medidores são thread safe e se redesenham na thread da interface; **OnValue** e **OnError** são executados na thread do cliente, portanto envie para a thread da interface antes de tocar em um controle.
- Um vínculo observa o evento **Disposed** de um controle WinForms e limpa **Gauge** quando o controle é descartado. Para controles WPF e Avalonia e para **GaugeModel**, limpe o vínculo antes de o medidor ser descartado.
- Uma string numérica é lida com '.' como separador decimal, qualquer que seja a cultura; true e false são lidos como 1 e 0.

A demonstração MQTTLive executa seis medidores sobre MQTT com um pequeno broker no próprio processo, veja Demos do .NET.

Veja também

- Visão geral da ponte
- Fonte MQTT
- Fonte WebSocket

- Fonte de eventos enviados pelo servidor
- Vinculação de dados no .NET
- Demos do .NET

Demos do .NET

As demonstrações .NET estão em `demos\Gauges` da pasta .NET do sgcUI, uma pasta por demonstração com um arquivo de projeto, Program.cs e um README.md. A interface é criada em código.

Demo	Host	Mostra
Gallery	WinForms	Os dez controles de medidor em um formulário: troca de tema, animação ligada e desligada, uma simulação de valor e exportação PNG com o dobro do tamanho. O limiar de velocidade em 200 km/h aciona um LED de alarme e o bloco HIGH SPEED; um clique em um bloco o reconhece.
GalleryWpf	WPF	A mesma galeria com os controles TsgcWpf.
GalleryAvalonia	Avalonia	A mesma galeria com os controles TsgcAvalonia.
HMIPanel	WinForms	Um processo simulado com dois tanques e duas bombas: medidores lineares de nível, medidores radiais de pressão, displays digitais de vazão, LEDs das bombas e um anunciador acionado por um único manipulador **OnThreshold** compartilhado.
MQTTLive	WinForms	Seis medidores alimentados por **TsgcGaugeFeedSource_MQTT**. Com a opção de loopback, Connect inicia um pequeno broker MQTT no processo, portanto a demonstração não precisa de broker nem de rede. Somente na edição Registered.
ConsoleRender	Console	Um modelo de cada tipo gravado em arquivos PNG e SVG, sem framework de interface; o equivalente da demonstração Headless do Delphi.

Compilar e executar

```
dotnet run --project Gallery\Gallery.csproj
```

Dentro da árvore de código do sgcUI, uma demonstração é compilada contra os projetos da biblioteca na edição Registered; adicione `-p:SgcCommunity=true` para a edição Community. No zip da demonstração, ela restaura os pacotes do nuget.org, por padrão os pacotes Community; defina `-p:SgcGaugesEdition=Registered` para os Registered. Na edição Community, as galerias mostram seis medidores, HMIPanel lista os alarmes em uma caixa de listagem no lugar do anunciador e ConsoleRender grava os seis tipos Community. MQTTLive precisa da edição Registered e do pacote esegece.sgcWebSockets.

Autoteste

Toda demonstração aceita `--test result.txt`: executa uma autoverificação fora da tela, grava uma linha PASS ou FAIL por verificação e termina com o número de verificações que falharam. `--snapshot file.png` salva uma imagem PNG e termina.

Veja também

- Visão geral do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Renderização sem janela no .NET
- Ponte sgcWebSockets no .NET

Solução de problemas

O valor para em 100

Um valor é limitado à faixa de sua escala quando é escrito, e a escala padrão vai de 0 a 100. Defina Minimum e Maximum da escala antes do valor. Uma mudança posterior da faixa limita novamente os valores armazenados, mas não restaura um valor que foi cortado.

A segunda escala ou ponteiro não é desenhado

A edição Community desenha apenas a primeira escala e o primeiro ponteiro. Os outros são mantidos nas coleções. Veja Edições.

TsgcVCLClockGauge ou TsgcCompassGauge não está declarado

O VU meter, o relógio, a bússola e o anunciador fazem parte da edição Registered. Na Community suas units estão vazias e seus controles não são registrados.

Um display de 14 segmentos mostra 7 segmentos

Os displays de 14 segmentos e de matriz de pontos fazem parte da edição Registered. Na Community são desenhados como 7 segmentos.

A agulha salta em vez de se mover

Verifique Animation.Enabled e Animation.Duration. Não há animação em tempo de projeto, para valores carregados do arquivo de formulário, para `SetValue(aPointer, aValue, False)` e em uma renderização sem janela, que desenha o valor de destino.

OnThreshold não dispara

- Com gtdRising e gtdFalling, o evento dispara apenas quando o limiar se torna ativo. Use gtdBoth para receber também o evento de término.
- Após um evento, o valor deve sair da banda dada por Hysteresis antes que o limiar possa disparar novamente.
- Uma mudança do próprio limiar atualiza Active sem evento.
- Verifique Enabled e PointerIndex. Na Community, os limiares dos ponteiros além do primeiro não são avaliados.

Nenhuma marca de limiar na escala

Um limiar só desenha uma marca quando seu Color não é 0.

Os rótulos mostram '.' em uma localidade com vírgula decimal

Os medidores usam seu próprio FormatSettings, com '.' por padrão, de modo que a saída não depende da máquina. Atribua FormatSettings do modelo, por meio da propriedade Gauge de um controle, para alterá-lo. Veja Temas.

O texto está pequeno ou grande demais

Os tamanhos de fonte são relativos ao tamanho do medidor, de modo que o texto acompanha o tamanho do controle. Altere o Size de TitleFont, ValueFont ou LabelFont para mudar a proporção.

Escrita de valores a partir de uma worker thread

Value, SetValue e o Value e o EndValue do ponteiro são thread safe, e os controles se repintam e chamam OnThreshold na thread principal. Todas as outras propriedades pertencem à thread principal.

Exemplo

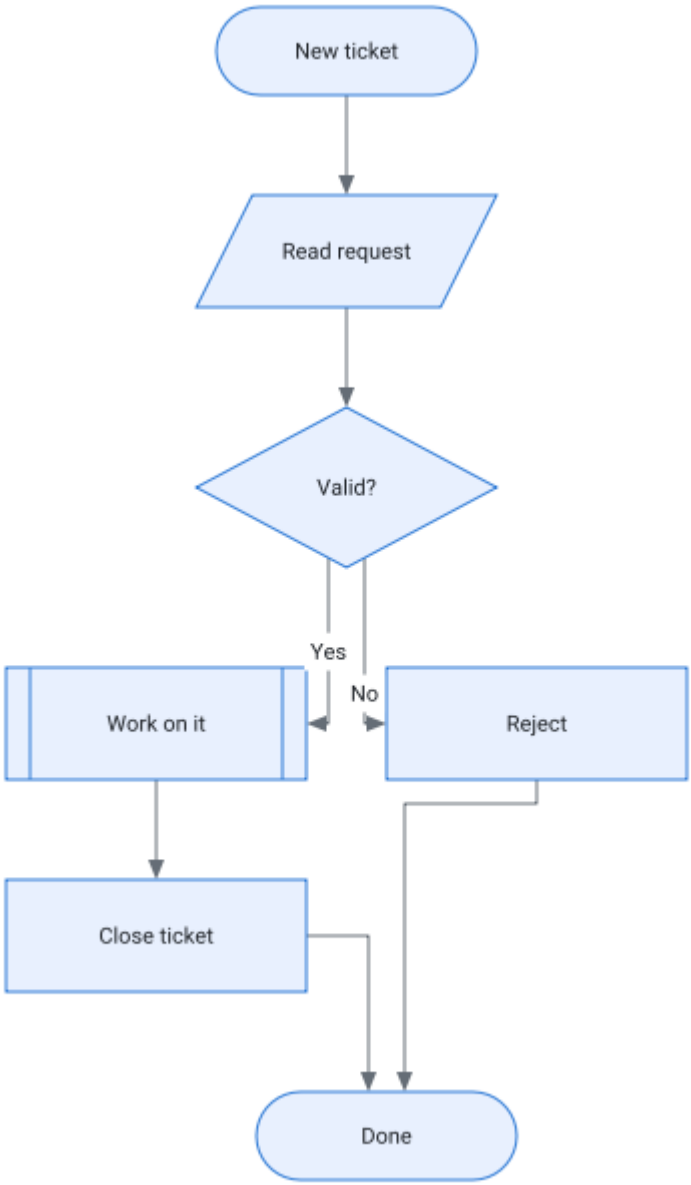
```
uses
  SysUtils, Classes, Controls, Forms,
  sgcGauges_Types, sgcGauges_Model, sgcGauges_VCL;

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;
```

Veja também

- Edições
- Animação
- Limiares

Introdução

sgcDiagram adiciona um editor e visualizador de diagramas a aplicativos Delphi e C++Builder, para VCL e FireMonkey: fluxogramas, organogramas e grafos de nós. O usuário arrasta formas de uma paleta, desenha conectores entre elas, move, redimensiona e gira os nós, edita o texto no próprio local e desfaz cada passo. O mesmo diagrama pode ser construído, organizado e exportado por código, sem janela, em um servidor ou no Linux64.
Tudo vive em um único objeto de documento, **TsgcDiagramDocument**: os nós, os conectores com suas portas e rótulos, as camadas, as imagens e os estilos padrão. **TsgcVCLDiagramView** e **TsgcFMXDiagramView** mostram e editam um documento, **TsgcVCLDiagramPalette** e **TsgcFMXDiagramPalette** oferecem as formas para arrastar até a view, e **TsgcDiagramRenderer** grava um documento em PNG e SVG sem uma view. Um documento é salvo e carregado como JSON, e uma view guarda seu diagrama no arquivo do formulário.
O sgcDiagram faz parte da suíte visual da eSeGeCe. Ele é desenhado com **sgcUICore**, o pacote de runtime compartilhado com o canvas, o rasterizador, as fontes, o gravador de SVG e os temas claro e escuro, de modo que um diagrama tem a mesma aparência no VCL, no FireMonkey e nos arquivos exportados. O sgcDiagram requer o sgcUICore, que é instalado junto com ele.



Edições

O sgcDiagram é distribuído como Registered, Trial e Community. A edição Community é um editor manual completo: todas as formas, caminhos personalizados, imagens, os três tipos de conector, portas e fixação, grupos e contêineres, desfazer, a área de transferência, JSON, PNG e SVG, temas, ajuste e guias, a paleta e o visualizador somente leitura. Ela deixa de fora os layouts automáticos, o roteamento ortogonal com desvio de obstáculos, o vínculo de dados e mais de uma camada. Veja Edições.

Por onde começar

- O que é o sgcDiagram: os tipos de diagrama que ele desenha e o que o usuário pode fazer com eles.
- Arquitetura: o documento, as views, o controlador e as units.
- Início Rápido VCL, Início Rápido FMX, Início Rápido C++Builder e Início Rápido Sem Janela: um primeiro diagrama em poucos minutos.
- Os componentes: TsgcVCLDiagramView, TsgcFMXDiagramView, TsgcVCLDiagramPalette, TsgcFMXDiagramPalette, TsgcDiagramDataBinding e seus Eventos.
- As classes: TsgcDiagramDocument, TsgcDiagramNode, TsgcDiagramConnector, TsgcDiagramPort, TsgcDiagramStyle, TsgcDiagramLayer, TsgcDiagramController, TsgcDiagramRenderer e o Formato JSON.
- Layout em Camadas, Layout em Árvore e Vínculo de Dados: organizam um diagrama automaticamente, ou o constroem a partir de um dataset.
- Os recursos: Formas, Conectores e Roteamento, Portas e Fixação, Edição e Teclado, Ajuste e Guias, Desfazer e Área de Transferência, Camadas, Serialização, Exportação, Temas e Modo Visualizador.
- Plataformas: as IDEs, os frameworks e os sistemas operacionais suportados.
- Solução de Problemas: os problemas comuns e suas causas.

O que é o sgcDiagram

O sgcDiagram é um conjunto de componentes de diagrama: um documento de nós e conectores, uma view interativa que o edita, uma paleta de formas e um exportador. Destina-se aos diagramas que os aplicativos de negócios mostram e deixam seus usuários editar.

O que você pode desenhar

- **Fluxogramas.** Formas de processo, decisão, terminador, dados, documento, processo predefinido e círculo conector, conectores com setas e rótulos Sim e Não.
- **Organogramas e árvores.** Caixas com um nome e uma função, unidas por conectores ortogonais. A edição Registered os organiza com um layout em árvore e os constrói a partir de um dataset.
- **Grafos de nós.** Nós com portas tipadas de entrada e saída, conectores curvos e seu próprio desenho em OnDrawNode, como em um editor de nós ou em um designer de pipelines.
- **Esboços livres.** Retângulos, retângulos arredondados, elipses, texto livre, imagens, formas personalizadas a partir de dados de caminho SVG e contêineres que guardam outros nós.

O que o usuário pode fazer

- Arrastar uma forma da paleta até a view, ou escolhê-la e desenhá-la com a ferramenta de forma.
- Selecionar com um clique, cliques com Shift e Ctrl ou um retângulo de seleção, depois mover, redimensionar, girar, ajustar com as setas, agrupar, excluir, copiar e colar.
- Desenhar um conector a partir de uma porta, fixar suas extremidades a portas ou ao contorno de um nó, mover seus rótulos e seus waypoints.
- Editar o texto de um nó no local: clique duplo, F2 ou simplesmente começar a digitar.
- Desfazer e refazer cada ação, rolar com a roda do mouse, o botão do meio ou Espaço, fazer zoom com Ctrl e a roda ou um gesto de pinça.

O que o código pode fazer

- Construir um diagrama com AddNode e AddConnector, alterar qualquer propriedade e agrupar as alterações em um passo de desfazer com BeginTransaction e EndTransaction.
- Salvá-lo e carregá-lo como JSON, como arquivo, stream ou string.
- Exportá-lo para PNG e SVG com TsgcDiagramRenderer, também em um aplicativo de console ou no Linux64.
- Ouvir OnChange, OnSelectionChange, OnItemClick, OnConnecting e os outros eventos para reagir ao usuário.

O que ele não faz

A versão 1 não tem stencils BPMN, UML ou de rede, nem pools de raias, nem layouts dirigidos por forças ou circulares, nem exportação para PDF, nem troca de área de transferência com outros aplicativos, nem minimapa, réguas, barras de rolagem ou impressão, nem importação de SVG ou Visio, nem texto rico nos nós.

Veja Também

- Arquitetura
- Edições
- Início Rápido VCL

Arquitetura

O sgcDiagram separa o documento, a interação e o desenho, de modo que o mesmo código edita um diagrama no VCL, no FireMonkey e em um teste sem janela.

Parte	Função
TsgcDiagramDocument	O modelo: Nodes, Connectors, Layers, Images, Selection, os NodeStyle e ConnectorStyle padrão, Routing, GridSize, Theme e a pilha de desfazer. Cada alteração é informada uma vez por par BeginUpdate e EndUpdate por meio de OnChange.
TsgcDiagramNode	Uma caixa com Shape, Left, Top, Width, Height, Angle, Text, Style, Ports, Layer, ContainerId, GroupId, ImageKey, PathData, Locked, Visible, Data e Tag. A ordem z é o índice em Nodes.
TsgcDiagramConnector	Uma ligação entre dois nós ou dois pontos livres: Kind, SourceNodeId, SourcePortId, TargetNodeId, TargetPortId, SourcePoint, TargetPoint, Waypoints, StartArrow, EndArrow, ArrowSize, Labels e Style. Points guarda a rota calculada.
Router	Calcula a rota de todo conector cujas extremidades, nós ou tipo mudaram, no fim de cada conjunto de alterações quando Routing.AutoRoute é True.
TsgcDiagramController	A máquina de estados da interação: teste de acerto, seleção, arraste, redimensionamento, rotação, conexão, ajuste, guias, edição de texto, zoom e rolagem. É independente de framework e trabalha em pixels do dispositivo por fora e em unidades de mundo por dentro.
TsgcDiagramPainter e o cache da view	Desenham o documento em um TsgcUICanvas do sgcUICore. A view mantém uma imagem em cache do conteúdo, de modo que uma rolagem é uma cópia de pixels e uma edição repinta apenas seu retângulo sujo.
TsgcVCLDiagramView, TsgcFMXDiagramView	Os controles. Eles possuem um documento (a propriedade Diagram) e um Controller, e traduzem o mouse, o teclado, os gestos e o arrastar e soltar em chamadas ao controlador.
TsgcVCLDiagramPalette, TsgcFMXDiagramPalette	Uma grade de itens de forma agrupados por categoria. Um clique escolhe a ferramenta de forma da view vinculada, um arraste solta a forma sobre ela.
TsgcDiagramRenderer	Exporta um documento para PNG e SVG sem uma view.

Coordenadas

As unidades de mundo são pixels em zoom 1 e 96 PPI, com y crescendo para baixo. Os ângulos dos nós estão em graus, no sentido horário, em torno do centro do nó. Uma cor é um TsgcUIColor, e 0 significa a cor do tema. ScreenToDiagram e DiagramToScreen da view convertem entre os pixels do dispositivo do controle e as unidades de mundo.

Notificações de alteração

O documento coleta as alterações entre o BeginUpdate e o EndUpdate mais externos e as informa uma vez: primeiro à view que o possui, que repinta o retângulo sujo, depois por OnChange com um conjunto de valores TsgcDiagramChange (dgkGeometry, dgkStyle, dgkStructure, dgkSelection, dgkView) e OnSelectionChange. Uma alteração feita pelo código fora de BeginUpdate e EndUpdate é informada de imediato.

Units

Unit	Conteúdo
sgcDiagram_Types	As enumerações, TsgcDiagramPoint, TsgcDiagramRect, os auxiliares de geometria como sgcDiagramPoint e sgcDiagramBounds, e EsgcDiagramError.
sgcDiagram_Shapes	O contorno, o retângulo de texto e as portas padrão de cada forma, e o analisador de dados de caminho SVG.
sgcDiagram_Model	TsgcDiagramDocument, os nós, conectores, portas, rótulos, camadas, imagens, estilos e a seleção.
sgcDiagram_Undo, sgcDiagram_JSON, sgcDiagram_Router	A pilha de desfazer, o formato JSON e a área de transferência, o roteamento de conectores.
sgcDiagram_Render	O painter, o cache da view e TsgcDiagramRenderer.
sgcDiagram_Palette, sgcDiagram_Controller	O modelo da paleta e o controlador de interação.
sgcDiagram_Layout, sgcDiagram_Data	Os layouts em camadas e em árvore e o vínculo de dados. Edição Registered; units vazias na Community.
sgcDiagram_VCL, sgcDiagram_FMX	A view e a paleta VCL e FireMonkey.

Veja Também

- O que é o sgcDiagram
- Plataformas
- Desfazer e Área de Transferência
- TsgcDiagramDocument
- TsgcDiagramController
- TsgcDiagramRenderer

Edições

O sgcDiagram é compilado a partir do mesmo código-fonte para todas as edições.

Edição	Inclui
Registered	Tudo: as views e paletas para VCL e FireMonkey, todas as formas, os três tipos de conector, portas, grupos e contêineres, desfazer, a área de transferência, JSON, PNG e SVG, temas, ajuste, guias e os recursos avançados listados abaixo. A página de paleta SGC Diagram mostra 5 componentes.
Trial	Os mesmos recursos da Registered. Uma compilação com SGC_TRIAL mostra uma mensagem de que é uma versão de avaliação quando um componente é criado depois que a janela de lançamento indicada na compilação passou. Nada mais muda.
Community	Tudo, exceto os recursos avançados: um editor manual completo de fluxogramas e organogramas para VCL, FireMonkey e uso sem janela. A página de paleta SGC Diagram mostra 4 componentes (as duas views e as duas paletas).

Os recursos avançados

Os recursos avançados são compilados sob a define SGC_DIAGRAM_ADVANCED, que o sgcVer.inc desativa na edição Community. As listas de units dos pacotes são as mesmas em todas as edições: sgcDiagram_Layout e sgcDiagram_Data são compiladas como units vazias na Community.

Recurso	Na edição Community
Layout automático: TsgcDiagramLayeredLayout e TsgcDiagramTreeLayout	Não disponível. Os nós permanecem onde o usuário ou o código os coloca.
Roteamento ortogonal com desvio de obstáculos e afastamento de segmentos paralelos	Os conectores dgcOrthogonal são roteados como cotovelos simples que não evitam os nós, e RouteFailed permanece False.
Vínculo de dados: TsgcDiagramDataBinding constrói um diagrama a partir de um dataset	Não disponível.
Camadas além da primeira	Layers.Add gera EsgcDiagramError quando já existe uma camada. Um arquivo salvo pela edição Registered mantém os nomes de suas camadas e é carregado e salvo sem alteração, mas toda camada é desenhada, exportada e editada como visível e desbloqueada.

Um arquivo de diagrama é o mesmo em todas as edições, de modo que um documento criado com a edição Registered abre em uma compilação Community e vice-versa.

Veja Também

- Conectores e Roteamento
- Camadas
- Plataformas
- Layout em Camadas
- Layout em Árvore
- TsgcDiagramDataBinding

Plataformas

O sgcDiagram é uma única árvore de código-fonte para todas as IDEs e plataformas. O documento, o roteador, o controlador, o painter e o renderizador usam apenas a RTL e o sgcUICore, de modo que são o mesmo código em todos os destinos; só as views e paletas estão ligadas a um framework.

Destino	Suporte
Delphi VCL	Delphi 7 ao Delphi 13, Win32 e Win64: TsgcVCLDiagramView e TsgcVCLDiagramPalette.
Delphi FireMonkey	Delphi XE6 ao Delphi 13 para Windows, macOS, iOS e Android: TsgcFMXDiagramView e TsgcFMXDiagramPalette. Compilado sob SGC_DIAGRAM_FMX.
Linux64	Delphi 10.2 e posterior, compilado sob SGC_DIAGRAM_LINUX: o documento, JSON, roteamento, layouts e TsgcDiagramRenderer, de modo que um servidor pode construir um diagrama e produzir arquivos PNG e SVG. Veja Início Rápido Sem Janela.
C++Builder	Os mesmos componentes com os cabeçalhos .hpp gerados. Veja Início Rápido C++Builder.
.NET	Um port em C# com a mesma API para WinForms, WPF e Avalonia, a view e a paleta em cada host e a saída headless de JSON, PNG e SVG, no .NET Framework 4.7.2 e no .NET 8 e 9, veja Visão geral do .NET.

Units por framework

Units	Framework
sgcDiagram_Const, sgcDiagram_Types, sgcDiagram_Shapes, sgcDiagram_Model, sgcDiagram_Undo, sgcDiagram_JSON, sgcDiagram_Router, sgcDiagram_Layout, sgcDiagram_Render, sgcDiagram_Palette, sgcDiagram_Controller	Nenhum. O modelo, os motores, o painter e o formato de arquivo.
sgcDiagram_Data	Somente a unit DB (dbrtl), nenhuma unit VCL ou FireMonkey.
sgcDiagram_VCL	VCL, somente Windows.
sgcDiagram_FMX	FireMonkey, Delphi XE6 e posterior.

Pacotes

O sgcDiagram traz um pacote de runtime independente de framework e um pacote de runtime por framework, de modo que um aplicativo FireMonkey para Android, iOS ou macOS nunca vincula a VCL. Cada nome de pacote termina com o sufixo do IDE, por exemplo sgcDiagramD13.dpk para o Delphi 13 e sgcDiagramC13.cbproj para o C++Builder 13.

Pacote	Tipo	Conteúdo	Exige
sgcDiagram	Runtime, independente	Toda unidade do produto que não usa nem a VCL nem o FireMonkey: o documento, as formas, o desfazer, JSON, o roteador, os layouts, o renderizador, o modelo da pale-	rtl, dbrtl, sgcUICore

	de framework	ta, o controlador e, fora da edição Community, TsgcDiagramDataBinding (sgcDiagram_Data).	
sgcDiagramVCL	Runtime, VCL	sgcDiagram_VCL (TsgcVCLDiagramView, TsgcVCLDiagramPalette)	rtl, vcl, sgcUICore, sgcUICoreVCL, sgcDiagram
sgcDiagramFMX	Runtime, FireMonkey, a partir do XE6	sgcDiagram_FMX (TsgcFMXDiagramView, TsgcFMXDiagramPalette)	rtl, fmx, sgcUICore, sgcUICoreFMX, sgcDiagram
dclsgcDiagram	Tempo de projeto, VCL	Registra TsgcVCLDiagramView, TsgcVCLDiagramPalette e, fora da edição Community, TsgcDiagramDataBinding na página **SGC Diagram** da paleta, e adiciona os verbos "Load Diagram...", "Save Diagram..." e "Clear Diagram" ao editor de componente da view.	designide, rtl, vcl, sgcDiagram, sgcDiagramVCL
dclsgcDiagramFMX	Tempo de projeto, FireMonkey, a partir do XE6	Registra TsgcFMXDiagramView e TsgcFMXDiagramPalette na página **SGC Diagram** da paleta, com os mesmos verbos.	designide, rtl, fmx, sgcDiagram, sgcDiagramFMX, dclsgcDiagram

O sgcUICore segue a mesma divisão: sgcUICore, sgcUICoreVCL e sgcUICoreFMX, sem pacote de tempo de projeto. O dclsgcDiagramFMX exige o dclsgcDiagram, portanto instale primeiro o dclsgcDiagram. O sgcDiagram não tem pacote de ponte com o sgcWebSockets.

Versões do IDE e plataformas de destino

Delphi	sgcDiagram	sgcDiagramVCL	sgcDiagramFMX	dclsgcDiagram, dclsgcDiagramFMX
7, 2007, 2009, 2010, XE	Win32	Win32	Nenhuma	Win32
XE2 a XE5	Win32, Win64	Win32, Win64	Nenhuma	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 ao 11	Win32, Win64	Igual ao 11	Win32, Win64

Do XE6 ao 10.2, o pacote FireMonkey é compilado apenas para Windows. Os pacotes de tempo de projeto são carregados pelo IDE, portanto são compilados apenas para a plataforma do IDE.

C++Builder	sgcDiagram, sgcDiagramVCL	sgcDiagramFMX	dclsgcDiagram, dclsgcDiagramFMX
2007, 2010, XE, XE2	Win32	Nenhuma	Win32 (somente dclsgcDiagram)
XE3 a XE5	Win32, Win64	Nenhuma	Win32 (somente dclsgcDiagram)
XE6 a 12	Win32, Win64	Win32, Win64	Win32
13	Win32, Win64, Win64x	Win32, Win64, Win64x	Win32, Win64x

Os projetos do C++Builder têm os mesmos nomes com o sufixo C, por exemplo sgcDiagramC13.cbproj, sgcDiagramVCLC13.cbproj, sgcDiagramFMXC13.cbproj, dclsgcDiagramC13.cbproj e dclsgcDiagramFMXC13.cbproj. Os pacotes do C++Builder são compilados apenas para Windows. As notas do C++Builder que valem para todos os produtos sgcUI (pacotes de runtime no Win64, C++Builder 10.3 no Win64, nomes de arquivo dos pacotes Win64x) estão no tópico Plataformas e Pacotes da ajuda da suíte sgcUI.

FireMonkey no Android, iOS e macOS

Os controles FireMonkey (TsgcFMXDiagramView e TsgcFMXDiagramPalette) têm como destino o macOS a partir do Delphi 10.3, o Android e o iOS a partir do Delphi 10.4, e o macOS ARM e o simulador do iOS a partir do Delphi 11. Em um projeto FireMonkey para essas plataformas:

- Use os controles FireMonkey. Os controles VCL e o sgcDiagramVCL funcionam apenas no Windows.
- Solte o controle em um formulário, ou adicione a unidade dele à cláusula uses: sgcDiagram_FMX.
- Selecione a plataforma de destino na janela Projetos. O instalador adiciona a pasta Source do sgcUI e as unidades compiladas de cada plataforma (LibD<version>\<platform>) ao caminho da biblioteca de cada plataforma que ele compila. Sem o instalador, adicione a pasta Source do sgcUI ao caminho da biblioteca da plataforma de destino (Ferramentas, Opções, Linguagem, Delphi, Biblioteca).
- Os aplicativos Android e iOS são vinculados estaticamente, portanto nenhum pacote de runtime é distribuído.

Linux64

O pacote independente de framework sgcDiagram é compilado para Linux64 a partir do Delphi 10.3. Um aplicativo Linux64, por exemplo um programa de console ou um servidor, usa o documento, JSON, o roteamento, os layouts e TsgcDiagramRenderer para montar um diagrama e produzir arquivos PNG e SVG. Nenhum pacote VCL ou FireMonkey é compilado para Linux64, porque o FireMonkey no Linux precisa do FMX para Linux (FMXLinux), um complemento de terceiros.

Notas por plataforma

- **Alto DPI.** A view VCL se adapta ao PPI do monitor e a view FireMonkey à escala do formulário. O zoom 1 mostra uma unidade de mundo por pixel independente de dispositivo.
- **Toque.** No FireMonkey, um gesto de pinça faz zoom em torno de seu centro e um dedo em espaço vazio faz a rolagem. Um dedo sobre um nó o arrasta, também em uma tela de toque do Windows.
- **Arrastar e soltar a partir da paleta.** No iOS e no Android não há serviço de arrastar: toque em um item da paleta, depois toque na view para posicionar a forma.
- **Tecla Tab.** Tab e Shift+Tab percorrem a seleção pelos itens no VCL. Um formulário FireMonkey trata Tab antes do controle, de modo que as teclas movem o foco ali.
- **Delphi 7 e Delphi 2007.** Essas versões têm strings Ansi, de modo que o texto fora da página de código do sistema é perdido quando um arquivo JSON é carregado.
- **Edições.** SGC_DIAGRAM_ADVANCED, que o sgcVer.inc desativa na edição Community, decide os recursos avançados em todas as plataformas. Veja Edições.

Veja Também

- Arquitetura
- Exportação
- Solução de Problemas

Início Rápido VCL

Instale os pacotes sgcDiagram para sua versão do Delphi. Os componentes aparecem na página de paleta **SGC Diagram**.

Em tempo de design

- Solte um **TsgcVCLDiagramPalette** no formulário e defina Align como alLeft.
- Solte um **TsgcVCLDiagramView** e defina Align como alClient.
- Defina a propriedade View da paleta como a view.
- Execute o aplicativo. Arraste formas da paleta, desenhe conectores a partir das portas que aparecem quando o ponteiro está sobre um nó, clique duas vezes em um nó para editar seu texto.

O menu de clique direito da view em tempo de design tem verbos para carregar um diagrama de um arquivo JSON, salvá-lo e limpá-lo. O diagrama é guardado no dfm como a propriedade DiagramJSON.

No 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 passos

- Salve o diagrama com FView.Diagram.SaveToFile, veja Serialização.

- Exporte-o com FView.SaveToPNG e FView.SaveToSVG, veja Exportação.
- Mude a view para um visualizador somente leitura com ReadOnly, veja Modo Visualizador.

Veja Também

- Início Rápido FMX
- Início Rápido Sem Janela
- Edição e Teclado
- TsgcVCLDiagramView
- TsgcVCLDiagramPalette

Início Rápido FMX

Os componentes FireMonkey são **TsgcFMXDiagramView** e **TsgcFMXDiagramPalette**, na página de paleta **SGC Diagram** do Delphi XE6 e posterior. Eles têm as mesmas propriedades, métodos e eventos dos componentes VCL, e a mesma classe de documento, de modo que o código que trabalha na propriedade Diagram é compartilhado.

Em tempo de design

- Solte um TsgcFMXDiagramPalette com Align Left e um TsgcFMXDiagramView com Align Client.
- Defina a propriedade View da paleta como a view.
- Execute o aplicativo no Windows ou no macOS e arraste formas para a view. No iOS e no Android, toque em um item da paleta, depois toque na view.

No 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;
```

Toque

- Um gesto de pinça faz zoom em torno de seu centro.
- Um dedo em espaço vazio faz a rolagem, um dedo sobre um nó o arrasta.

Veja Também

- Início Rápido VCL
- Plataformas
- TsgcFMXDiagramView

- TsgcFMXDiagramPalette

Início Rápido C++Builder

Instale os pacotes sgcDiagram para sua versão do C++Builder, adicione `sgcUI\Source` e a pasta das units compiladas, por exemplo `sgcUI\LibD13\win32` (`sgcUI\LibD13\win64` para 64 bits), aos caminhos de include e de biblioteca do projeto, e use os componentes da página de paleta **SGC Diagram** exatamente como no Delphi.

O que saber em C++

- As enumerações não têm escopo: escreva `dgsDecision`, `dgcCurved` ou `dgtSelect`.
- Os exemplos desta ajuda passam todos os argumentos de um método explicitamente, por exemplo `AddConnector(Start, Check, "", "")`, de modo que compilam com qualquer versão dos cabeçalhos gerados.
- Os eventos são closures, de modo que seus manipuladores são funções membro __fastcall de uma classe. Um parâmetro var como aAccept é uma referência bool.
- TsgcDiagramRenderer é uma classe simples: crie-a com `new TsgcDiagramRenderer()` e a exclua com delete.

No código

Veja Também

- Início Rápido VCL
- Plataformas

Início Rápido Sem Janela

Um diagrama não precisa de uma view. TsgcDiagramDocument e TsgcDiagramRenderer usam apenas a RTL e o sgcUICore, de modo que um aplicativo de console, um serviço ou um servidor Linux64 pode construir um diagrama, roteá-lo e gravar arquivos PNG, SVG e JSON.

- Crie o documento com `TsgcDiagramDocument.Create` (sem proprietário) e libere-o você mesmo.
- Os conectores são roteados no fim de cada conjunto de alterações enquanto Routing.AutoRoute é True (o padrão). Chame RouteAll para rotear todos os conectores novamente.
- O renderizador exporta os limites do conteúdo mais uma margem, 20 unidades de mundo por padrão, na escala que você passar.

Exemplo

```
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;
```

Veja Também

- Exportação
- Serialização
- Plataformas
- TsgcDiagramDocument
- TsgcDiagramRenderer
- Layout em Camadas

TsgcVCLDiagramView

TsgcVCLDiagramView (unit sgcDiagram_VCL) é o controle VCL que mostra e edita um diagrama. Ele possui um TsgcDiagramDocument, publicado como Diagram, e um TsgcDiagramController que trata o mouse, o teclado e o arrastar e soltar.

Propriedades

Propriedade	Significado
Diagram	O TsgcDiagramDocument mostrado e editado pela view. É gravado em stream no arquivo do formulário como a propriedade DiagramJSON, escrita somente quando o diagrama não está vazio.
Tool	A ferramenta de ponteiro: dgtSelect (padrão), dgtPan, dgtConnector ou dgtShape.
ShapeTool	A forma desenhada pela ferramenta dgtShape, dgsRectangle por padrão.
ReadOnly	True transforma a view em um visualizador: seleção, hover, zoom e rolagem funcionam, a edição não.
Zoom, MinZoom, MaxZoom	O fator de zoom e seus limites (0.05 e 8 por padrão).
ShowGrid	Desenha a grade de Diagram.GridSize (True por padrão).
SnapToGrid, SnapToGuides	Ajusta os nós movidos e redimensionados à grade e às bordas, centros e espaços iguais dos nós ao redor (ambos True por padrão).
ShowPorts	Quando as portas dos nós são desenhadas: dgvNever, dgvHover (padrão) ou dgvAlways.
Theme	O tema do diagrama, o mesmo valor de Diagram.Theme (não é guardado separadamente).
FontFile	Um arquivo de fonte usado para desenhar o texto, a mesma fonte na view e na exportação.
MaxFPS	A maior taxa de repintura do controle.
Controller	Pública. O TsgcDiagramController que transforma o ponteiro e o teclado em edições.

Métodos

Método	Efeito
ZoomToFit, ZoomToSelection	Aplicam zoom para que todo o diagrama, ou a seleção, preencha a view.
CenterOn(aPoint)	Rola um ponto de mundo para o centro da view.
ScreenToDiagram, DiagramToScreen	Convertem entre os pixels do dispositivo da view e as unidades de mundo.
Undo, Redo	Desfazem ou refazem o último passo do documento.

CopyToClipboard, CutToClipboard, PasteFromClipboard	A área de transferência do diagrama, veja Desfazer e Área de Transferência.
DeleteSelection	Exclui os itens selecionados como um único passo de desfazer.
EditText(aNode)	Abre o editor no local sobre o texto de um nó; retorna False quando a edição não é permitida.
SaveToPNG, SaveToSVG	Exportam para um arquivo ou um stream, com uma escala opcional (1 por padrão).

Eventos

A view publica OnChange, OnSelectionChange, OnItemClick, OnItemDblClick, OnHover, OnConnecting, OnConnected, OnDrawNode, OnTextEditing, OnTextEdited e OnZoomChange, descritos em Eventos.

Notas

- A view não tem barras de rolagem. A roda rola, Shift e a roda rolam na horizontal, Ctrl e a roda fazem zoom em torno do ponteiro, e o botão do meio, Espaço ou a ferramenta dgtPan fazem a rolagem, veja Edição e Teclado.
- O editor de texto é um memo no local: Enter ou sair do memo confirma, Esc cancela e Shift+Enter adiciona uma linha.
- Soltar um item de TsgcVCLDiagramPalette cria o nó, ou o fragmento, no ponto de soltura.

Exemplo

```
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;
```

Veja Também

- TsgcFMXDiagramView
- TsgcVCLDiagramPalette
- TsgcDiagramDocument
- Início Rápido VCL

TsgcFMXDiagramView

TsgcFMXDiagramView (unit sgcDiagram_FMX) é o controle FireMonkey que mostra e edita um diagrama no Windows, macOS, Linux, iOS e Android. Ele tem as mesmas propriedades, métodos e eventos de TsgcVCLDiagramView: Diagram, Tool, ShapeTool, ReadOnly, Zoom, MinZoom, MaxZoom, ShowGrid, SnapToGrid, SnapToGuides, ShowPorts, Theme, FontFile, MaxFPS, Controller, os métodos de zoom, área de transferência, desfazer e exportação, e OnChange, OnSelectionChange, OnItemClick, OnItemDblClick, OnHover, OnConnecting, OnConnected, OnDrawNode, OnTextEditing, OnTextEdited e OnZoomChange, descritos em Eventos.

Diferenças em relação à view VCL

- O diagrama é gravado em stream no arquivo fmx como a propriedade DiagramJSON, escrita somente quando o diagrama não está vazio.
- As posições estão em pixels do dispositivo: as coordenadas locais do controle vezes sua escala.
- Um gesto de pinça faz zoom em torno do centro da pinça. Um dedo que começa sobre um item o arrasta; um dedo que começa em espaço vazio faz a rolagem.
- A roda do mouse do FMX não tem posição, de modo que a roda faz zoom em torno da última posição do ponteiro.
- O iOS e o Android não têm serviço de arrastar e soltar: toque em um item da paleta para armá-lo, depois toque na view para posicionar a forma.
- O formulário FMX trata Tab antes do controle com foco, de modo que Tab e Shift+Tab não percorrem os itens.
- O editor de texto é um TMemo FMX simples que segue o estilo do formulário.

Exemplo

```
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;
```

Veja Também

- TsgcVCLDiagramView
- TsgcFMXDiagramPalette
- Início Rápido FMX
- Modo Visualizador

TsgcVCLDiagramPalette

TsgcVCLDiagramPalette (unit sgcDiagram_VCL) mostra as formas que o usuário pode adicionar a um TsgcVCLDiagramView, por clique ou por arrastar e soltar.

Propriedades

Propriedade	Significado
View	A view de diagrama que recebe os itens clicados ou arrastados.
Items	Os itens, uma coleção de TsgcDiagramPaletteItem mostrada em uma grade, agrupada por Category.
IconSize	O tamanho do ícone de um item em pixels, 32 por padrão.
ShowCaptions	Desenha a legenda sob cada ícone (True por padrão).
Theme	utLight (padrão), utDark ou utCustom.
MaxFPS	A maior taxa de repintura do controle.
Model	Pública. O modelo da paleta por trás do controle: layout, teste de acerto e payload de arraste.
DragPayload	Pública. O payload do arraste em andamento: o JSON do item arrastado.

Itens

TsgcDiagramPaletteItem	Significado
Category, Caption	O cabeçalho do grupo e o texto sob o ícone.
Shape, Width, Height, Text, Style, PathData	O nó que o item cria (PathData para dgsPath, em uma caixa de 0..100).
Fragment	Um fragmento JSON, como retornado por Diagram.CopySelection. Um item com fragmento solta o fragmento inteiro.
Payload	Somente leitura. O JSON solto pelo item.

Eventos

Evento	Quando
OnItemClick(Sender; aItem)	Um item recebe um clique.

Notas

- Um clique em um item define o Tool da view como dgtShape e seu ShapeTool como a forma do item; um clique ou um arraste na view então cria o nó, e Esc volta para dgtSelect. Um item de fragmento só é posicionado por arrastar e soltar.
- Arrastar um item para a view mostra uma prévia e solta o nó, ou o fragmento, no ponteiro como um único passo de desfazer. O payload nunca passa pela área de transferência do sistema.
- Items.AddShape(aCategory, aCaption, aShape, aWidth, aHeight, aText) e Items.AddFragment(aCategory, aCaption, aFragment) adicionam itens no código. LoadDefaultItems restaura os itens padrão.

Exemplo

```
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;
```

Veja Também

- TsgcFMXDiagramPalette
- TsgcVCLDiagramView
- Formas

TsgcFMXDiagramPalette

TsgcFMXDiagramPalette (unit sgcDiagram_FMX) é a paleta FireMonkey para um TsgcFMXDiagramView. Ela tem as propriedades, os itens e os eventos de TsgcVCLDiagramPalette: View, Items, IconSize, ShowCaptions, Theme, MaxFPS, Model, DragPayload, OnItemClick e LoadDefaultItems.

Diferenças em relação à paleta VCL

- Nas plataformas desktop, um arraste inicia o arrastar e soltar FMX do formulário; a view mostra a prévia enquanto o ponteiro está sobre ela.
- No iOS e no Android não há serviço de arrastar: um toque em um item arma a ferramenta de forma da view, e um toque na view posiciona a forma.

Notas

- Um clique em um item define o Tool da view como dgtShape e seu ShapeTool como a forma do item; um clique ou um arraste na view então cria o nó, e Esc volta para dgtSelect. Um item de fragmento só é posicionado por arrastar e soltar.
- Arrastar um item para a view mostra uma prévia e solta o nó, ou o fragmento, no ponteiro como um único passo de desfazer. O payload nunca passa pela área de transferência do sistema.
- Items.AddShape(aCategory, aCaption, aShape, aWidth, aHeight, aText) e Items.AddFragment(aCategory, aCaption, aFragment) adicionam itens no código. LoadDefaultItems restaura os itens padrão.

Exemplo

```
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;
```

Veja Também

- TsgcFMXDiagramView
- TsgcVCLDiagramPalette

TsgcDiagramDataBinding

TsgcDiagramDataBinding (unit sgcDiagram_Data) constrói um diagrama a partir de linhas, tipicamente um organograma ou um fluxograma: cada linha vira um nó, e cada chave de pai vira um conector do pai para o filho. É independente de framework e funciona com a view VCL, a view FMX ou um documento sozinho. Os nomes de campos e colunas não diferenciam maiúsculas de minúsculas.

Propriedades

Propriedade	Significado
DataSource	O TDataSource cujo DataSet fornece as linhas.
Table	Pública. Uma IsgcUITable que fornece as linhas; quando atribuída, é lida em vez de DataSource. Defina como nil antes de o objeto de tabela ser liberado.
KeyField	O campo com a chave única de uma linha. A chave se torna o Id do nó, de modo que Document.FindNode(key) encontra o nó.
ParentField	O campo com a chave do pai, ou várias chaves separadas por ParentSeparator.
ParentSeparator	O separador de várias chaves de pai, um ponto e vírgula por padrão. Vários pais formam um fluxograma em vez de uma árvore.
LabelField	O campo mostrado como texto do nó; a chave quando vazio.
Target	A view que mostra o diagrama: qualquer componente que implemente IsgcDiagramDocumentOwner, como TsgcVCLDiagramView e TsgcFMXDiagramView.
Document	Pública. Um documento a preencher diretamente; quando atribuído, é usado em vez do documento de Target. Não pertence ao vínculo: defina como nil antes de o documento ser liberado.
Layout	dgbAuto (padrão) aplica o layout em árvore quando nenhum nó tem dois pais, senão o layout em camadas; dgbTree e dgbLayered forçam um deles; dgbNone mantém os nós em linhas, na ordem das linhas.
Direction, TreeStyle	Passados ao layout: dglTopToBottom e dgoStandard por padrão.
NodeShape, NodeWidth, NodeHeight	A forma e o tamanho dos nós: dgsRoundRect, 140 por 60 por padrão.
ConnectorKind	O tipo dos conectores, dgcOrthogonal por padrão.
LiveUpdates	True (padrão) reconstrói o diagrama depois que os dados mudam.
Active	True constrói o diagrama e começa a acompanhar os dados.

Métodos e eventos

Membro	Significado

Refresh	Reconstrói o diagrama agora. Gera EsgcDiagramError dentro de uma transação do documento.
OnCreateNode(Sender; aNode; aRow)	Executado para cada nó depois de definidos seu texto, forma e tamanho. aRow é a linha de base 0, e o DataSet está posicionado nela.
OnCreateConnector(Sender; aConnector; aRow)	Executado para cada conector quando todos os nós existem. aRow é a linha do nó filho; o DataSet não está posicionado nela.

Regras

- Uma linha com chave vazia, ou com uma chave que uma linha anterior usou, não gera nó. Uma chave de pai vazia, desconhecida ou repetida, ou a própria chave da linha, não gera conector, de modo que uma linha cujos pais são todos desconhecidos é uma raiz.
- Uma reconstrução limpa o documento (itens, camadas e imagens), cria os itens, aplica o layout e roteia os conectores em um único BeginUpdate, de modo que a view é repintada uma vez. Não é um passo de desfazer e limpa as duas pilhas de desfazer.
- O vínculo é de mão única: ele nunca grava nos dados, e as alterações feitas no diagrama são substituídas pela próxima reconstrução.
- Nenhuma reconstrução é executada em tempo de design, de modo que as views nunca guardam um diagrama gerado no arquivo do formulário.
- Veja Vínculo de Dados para atualizações ao vivo e um exemplo completo.

Edição Community

TsgcDiagramDataBinding requer SGC_DIAGRAM_ADVANCED: na edição Community a unit sgcDiagram_Data está vazia e o componente não é registrado.

Exemplo

```
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;
```

Veja Também

- Vínculo de Dados
- Layout em Árvore
- Layout em Camadas
- Edições

Eventos

Os eventos das views vêm de seu controlador; o documento e o vínculo têm os seus próprios. As views VCL e FMX publicam os mesmos eventos com os mesmos parâmetros.

Eventos da view

Evento	Quando
OnChange(Sender; aChanges)	O documento mudou. aChanges é um conjunto de dgkGeometry, dgkStyle, dgkStructure, dgkSelection e dgkView.
OnSelectionChange(Sender)	A seleção mudou.
OnItemClick(Sender; aItem), OnItemDblClick(Sender; aItem)	Um nó ou conector recebe um clique ou um clique duplo.
OnHover(Sender; aItem)	O ponteiro entra em um item, ou sai dele (aItem é nil).
OnConnecting(Sender; aSource, aSourcePort, aTarget, aTargetPort; var aAccept)	Um arraste de conector pergunta antes de fixar uma extremidade. Uma porta vazia é o contorno do nó, um nó nil é uma extremidade livre. Defina aAccept como False para recusar.
OnConnected(Sender; aItem)	Um conector foi criado ou reconectado pelo usuário.
OnDrawNode(Sender; aCanvas; aNode; aRect; aZoom; aScale; var aDefault)	Antes de um nó ser pintado. aRect é a caixa sem rotação em unidades de mundo, e a transformação do canvas mapeia unidades de mundo em pixels do dispositivo com a rotação do nó aplicada; uma unidade de mundo equivale a aZoom x aScale pixels do dispositivo. Desenhe em aCanvas, um TsgcUICanvas, e defina aDefault como False para pular a pintura padrão da forma, da imagem e do texto. As legendas das portas são pintadas de qualquer forma.
OnTextEditing(Sender; aNode; var aAllow)	O editor de texto está prestes a abrir. Defina aAllow como False para manter o texto somente leitura.
OnTextEdited(Sender; aNode)	O texto de um nó foi alterado pelo editor.
OnZoomChange(Sender)	O zoom mudou.

Outros eventos

Evento	Quando
TsgcDiagramDocument.OnChange, OnSelectionChange	As mesmas notificações, para um documento sem view.
TsgcDiagramDocument.OnDecodeImage(Sender; aKey; aStream; var aImage; var aHandled)	Uma imagem em um formato diferente de PNG é carregada: decodifique aStream em aImage e defina aHandled.
TsgcDiagramDataBinding.OnCreateNode, OnCreateConnector	Veja TsgcDiagramDataBinding.

TsgcVCLDiagramPalette.OnItemClick, TsgcFMXDiagramPalette.OnItemClick	Um item da paleta recebe um clique.

Exemplo

```
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;
```

Veja Também

- TsgcVCLDiagramView
- TsgcDiagramController
- Portas e Fixação

TsgcDiagramDocument

TsgcDiagramDocument (unit sgcDiagram_Model) guarda um diagrama: seus nós, conectores, camadas e imagens, a seleção e o histórico de desfazer. Cada view mostra um documento, publicado como sua propriedade Diagram, e um documento também funciona sozinho, sem view, em um aplicativo de console ou em um serviço.

Propriedades

Propriedade	Significado
Nodes, Connectors	Os itens, em ordem z (o último fica no topo).
Layers, Images	As camadas e as imagens mostradas pelos nós de imagem, por chave (Images.Add, AddPNG, Find, Delete).
Selection	Os itens selecionados: Add, Remove, Toggle, Clear, SelectAll, Contains, Count e Items.
Theme	utLight (padrão), utDark ou utCustom.
GridSize	O passo da grade em unidades de mundo, 10 por padrão.
NodeStyle, ConnectorStyle	Os estilos usados quando um item deixa um valor em 0.
DefaultConnectorKind	O tipo dos novos conectores, dgcOrthogonal por padrão.
Routing	As configurações do roteador: Margin (12), BendPenalty (40), Spacing (6) e AutoRoute (True).
UndoLimit	O número de passos de desfazer mantidos, 100 por padrão.
InTransaction, Loading	Flags de estado somente leitura.

Métodos

Método	Efeito
AddNode(aShape, aX, aY, aWidth, aHeight, aText)	Adiciona um nó e o retorna.
AddConnector(aSource, aTarget, aSourcePort, aTargetPort)	Adiciona um conector entre dois nós; uma porta vazia fixa a extremidade ao contorno.
FindNode(aId), FindConnector(aId)	Encontram um item pelo Id, nil quando não é encontrado.
ConnectorsOf(aNode, aList), ChildrenOf(aContainer, aList)	Preenchem uma TList com os conectores de um nó, ou os filhos de um contêiner.
ItemsInRect(aRect, aList, aFullyInside)	Preenche uma TList com os itens que tocam, ou ficam dentro de, um retângulo.
ContentBounds	Os limites de todo item visível.

DeleteSelection, Clear	Excluem os itens selecionados, ou tudo.
Group, Ungroup, BringToFront, SendToBack	Atuam sobre a seleção. Group retorna o novo Id do grupo.
BeginUpdate, EndUpdate	Uma repintura e uma notificação de alteração para muitas alterações.
BeginTransaction(aCaption), EndTransaction, Rollback	Muitas alterações como um único passo de desfazer, ou nenhuma delas.
BeginUndoSuspend, EndUndoSuspend	As alterações intermediárias não são registradas e limpam as duas pilhas de desfazer; para cargas em massa.
Undo, Redo, CanUndo, CanRedo, UndoCaption, RedoCaption	O histórico de desfazer.
RouteAll	Roteia todos os conectores novamente.
SaveToJSON, LoadFromJSON, SaveToFile, LoadFromFile, SaveToStream, LoadFromStream	O formato JSON.
CopySelection, Paste(aJSON)	Um fragmento do documento como JSON, e sua colagem.
ResolveNodeStyle(aNode), ResolveConnectorStyle(aConnector)	O estilo com o qual um item é desenhado, com as cores do tema resolvidas.
LayerVisible(aName), LayerLocked(aName)	Como uma camada é tratada.

Eventos

OnChange(Sender; aChanges), OnSelectionChange e OnDecodeImage, veja Eventos.

Threads

Um documento é alterado somente na thread que o possui, a thread principal quando é mostrado em uma view.

Exemplo

```
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;
```

Veja Também

- TsgcDiagramNode
- TsgcDiagramConnector
- Desfazer e Área de Transferência
- Arquitetura

TsgcDiagramNode

Um nó é uma forma com um texto: uma etapa de processo, uma decisão, uma pessoa de um organograma ou um contêiner de outros nós. Os nós são criados com Document.AddNode e ficam em Document.Nodes.

Propriedades

Propriedade	Significado
Shape	A forma, dgsRectangle por padrão.
Left, Top, Width, Height	A caixa em unidades de mundo (120 por 60 por padrão).
Angle	A rotação em graus, no sentido horário, em torno do centro.
Text, TextAlign, TextVAlign	O texto e seu alinhamento (centralizado por padrão).
Ports	As portas; vazio usa as portas padrão da forma.
ContainerId	O Id do nó contêiner que guarda este nó.
GroupId	O grupo do nó; os nós de um grupo são selecionados e movidos juntos.
ImageKey	A chave em Document.Images da imagem desenhada por um nó dgsImage.
PathData	O contorno de um nó dgsPath, na sintaxe de caminho SVG em uma caixa de 0..100.
Visible	False oculta o nó.
Id	O Id único do item no documento; AddNode e AddConnector geram um.
Style	O estilo do item; os valores em 0 vêm do estilo do documento.
Layer	O nome da camada; vazio para a camada padrão.
Locked	True impede que o item seja selecionado e editado.
Data, Tag	Dados da aplicação: uma string (salva no arquivo) e um inteiro.
Selected, Document	Pública, somente leitura.

Métodos

Método	Efeito
SetBounds(aLeft, aTop, aWidth, aHeight), MoveBy(aDX, aDY)	Movem ou redimensionam o nó em uma única alteração.
BoundsRect, Center	A caixa e seu centro.
PortPoint(aPortId)	A posição de mundo de uma porta.

Exemplo

```
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;
```

Veja Também

- TsgcDiagramPort
- TsgcDiagramConnector
- Formas

TsgcDiagramConnector

Um conector une dois nós, ou um nó e um ponto livre. Sua rota é calculada pelo roteador a partir de seu Kind. Os conectores são criados com Document.AddConnector e ficam em Document.Connectors.

Propriedades

Propriedade	Significado
Kind	dgcStraight, dgcOrthogonal ou dgcCurved.
SourceNodeId, SourcePortId, TargetNodeId, TargetPortId	As extremidades fixadas. Uma porta vazia fixa a extremidade ao contorno do nó.
SourcePoint, TargetPoint	As extremidades livres, usadas quando o Id do nó está vazio.
Waypoints	Os pontos pelos quais um conector reto ou curvo passa.
StartArrow, EndArrow, ArrowSize	As pontas de seta (dgaNone e dgaTriangle por padrão) e seu tamanho (10).
Labels	Os rótulos: Text, Position ao longo do caminho de 0 a 1 (0.5 por padrão) e Offset em relação ao caminho.
Points	Pública, somente leitura. A rota atual.
RouteFailed	Pública, somente leitura. True quando o roteador não encontrou nenhum caminho ao redor dos nós.
Id	O Id único do item no documento; AddNode e AddConnector geram um.
Style	O estilo do item; os valores em 0 vêm do estilo do documento.
Layer	O nome da camada; vazio para a camada padrão.
Locked	True impede que o item seja selecionado e editado.
Data, Tag	Dados da aplicação: uma string (salva no arquivo) e um inteiro.
Selected, Document	Pública, somente leitura.

Métodos

Método	Efeito
SetRoute(aPoints, aFailed)	Define a rota; usado pelo roteador e pelos layouts.

Edição Community

Sem SGC_DIAGRAM_ADVANCED, um conector dgcOrthogonal é roteado como um cotovelo: sem desvio de obstáculos e sem afastamento, e RouteFailed permanece False.

Exemplo

```
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;
```

Veja Também

- Conectores e Roteamento
- TsgcDiagramPort
- TsgcDiagramNode

TsgcDiagramPort

Uma porta é um ponto nomeado em um nó onde as extremidades dos conectores são fixadas. Um nó com uma coleção Ports vazia usa as portas padrão de sua forma: os pontos médios dos lados, chamados left, top, right e bottom.

Propriedades

Propriedade	Significado
Id	O nome usado por SourcePortId e TargetPortId.
X, Y	A posição como fração da caixa do nó, de 0 a 1 (0.5 por padrão).
Side	O lado de onde sai um conector: dgpAuto (padrão), dgpLeft, dgpTop, dgpRight ou dgpBottom.
Flow	dgfBoth (padrão), dgfIn ou dgfOut: quais extremidades podem se fixar à porta.
Caption	Um texto desenhado ao lado da porta.

Node.Ports.Add adiciona uma porta e Node.Ports.Find(aId) encontra uma. Node.PortPoint(aId) retorna sua posição de mundo.

Exemplo

```
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;
```

Veja Também

- Portas e Fixação
- TsgcDiagramNode

TsgcDiagramStyle

Todo nó e conector tem um Style, e o documento tem um NodeStyle e um ConnectorStyle. Uma cor ou um tamanho deixado em 0 vem do estilo do documento, e uma cor ainda em 0 ali vem do tema, de modo que um diagrama acompanha uma mudança de tema, a menos que suas cores sejam definidas.

Propriedades

Propriedade	Significado
Fill	A cor de preenchimento, um TsgcUIColor (crie um com sgcUIARGB).
Stroke	A cor do contorno ou da linha.
StrokeWidth	A largura do contorno ou da linha em unidades de mundo, 1 por padrão.
Dash	dgdSolid (padrão), dgdDash ou dgdDot.
CornerRadius	O raio dos cantos de retângulos e retângulos arredondados.
Text	O estilo do texto: Size, Bold e Color.

Document.ResolveNodeStyle(aNode) e ResolveConnectorStyle(aConnector) retornam os valores com os quais um item é desenhado.

Exemplo

```
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;
```

Veja Também

- Temas
- TsgcDiagramDocument

TsgcDiagramLayer

Uma camada agrupa itens por nome. Document.Layers guarda as camadas; cada nó e conector indica sua camada na propriedade Layer, e um nome vazio é a camada padrão.

Propriedades

Propriedade	Significado
Name	O nome usado pela propriedade Layer dos itens.
Visible	False oculta os itens da camada na view e na exportação.
Locked	True impede que os itens da camada sejam selecionados e editados.

Layers.Add adiciona uma camada e Layers.Find(aName) encontra uma. Adicionar, excluir e renomear uma camada são passos de desfazer; Visible e Locked não são.

Edição Community

Sem SGC_DIAGRAM_ADVANCED, um segundo Layers.Add gera EsgcDiagramError. Um arquivo com mais camadas é carregado e salvo sem alteração, mas toda camada é desenhada e editada como visível e desbloqueada.

Exemplo

```
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;
```

Veja Também

- Camadas
- Edições

TsgcDiagramController

TsgcDiagramController (unit sgcDiagram_Controller) é o motor de edição independente de framework das views: ele transforma a entrada de ponteiro, teclado, roda do mouse e arrastar e soltar em alterações do documento, mantém o zoom e a posição de rolagem, e desenha por meio de um TsgcDiagramPainter. Cada view cria um e o expõe como sua propriedade Controller. As posições estão em pixels do dispositivo; internamente o controlador trabalha em unidades de mundo.

Propriedades

Propriedade	Significado
Document, Host, Painter	O documento, o controle hospedeiro (IsgcDiagramHost) e o painter.
Tool, ShapeTool, ShapeWidth, ShapeHeight	A ferramenta ativa e a forma e o tamanho padrão que a ferramenta de forma desenha.
ReadOnly, ShowGrid, SnapToGrid, SnapToGuides, ShowPorts	As opções publicadas pelas views.
Zoom, MinZoom, MaxZoom, OffsetX, OffsetY, Viewport	A transformação da view.
State	A interação em andamento: dgiIdle, dgiMoving, dgiResizing e assim por diante.
HoverItem, EditingNode	O item sob o ponteiro e o nó cujo texto está sendo editado.

Métodos

Método	Efeito
ZoomAt(aZoom, aX, aY), ScrollBy(aDX, aDY)	Zoom em torno de um ponto do dispositivo, rolagem em pixels do dispositivo.
ZoomToFit, ZoomToSelection, CenterOn	Como nas views.
SelectAll, Duplicate, Nudge(aDX, aDY)	Seleciona tudo, duplica a seleção, move-a em unidades de mundo.
GroupSelection, UngroupSelection	Agrupa ou desagrupa os nós selecionados.
Undo, Redo, CopyToClipboard, CutToClipboard, PasteFromClipboard, DeleteSelection	Os comandos de edição.
EditText(aNode), EndTextEdit	Abre e fecha o editor de texto.
Cancel	Cancela a interação em andamento, como faz Esc.

Os eventos do controlador são publicados pelas views, veja Eventos. O teclado é descrito em Edição e Teclado.

Exemplo

```
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;
```

Veja Também

- TsgcVCLDiagramView
- Edição e Teclado

TsgcDiagramRenderer

TsgcDiagramRenderer (unit sgcDiagram_Render) desenha um documento sem uma view: em arquivos ou streams PNG e SVG, ou em qualquer TsgcUICanvas. É o que as views usam para SaveToPNG e SaveToSVG, e funciona em aplicativos de console, em serviços e no Linux.

Métodos

Método	Efeito
SaveToPNGFile(aDocument, aFileName, aScale, aMargin), SaveToPNGStream	Um PNG de todo o diagrama. aScale é 1 por padrão, aMargin 20 unidades de mundo.
SaveToSVGFile(aDocument, aFileName, aScale, aMargin), SaveToSVGStream	O mesmo como SVG.
ExportViewport(aDocument, aScale, aMargin)	A viewport que uma exportação usa: deslocamento, zoom, escala e tamanho da imagem.
RenderTo(aCanvas, aDocument, aViewport)	Desenha o documento em um canvas por meio de uma viewport.
Painter	O TsgcDiagramPainter que desenha; seu evento OnDrawNode personaliza os nós também na exportação.

Exemplo

```
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;
```

Veja Também

- Exportação
- Início Rápido Sem Janela

Formato JSON

SaveToJSON grava um único objeto JSON. A saída é canônica: uma ordem de chaves fixa, sem indentação, e todo valor igual ao padrão é omitido, de modo que uma chave ausente sempre significa o padrão. As enumerações são gravadas como nomes, as cores como strings hexadecimais, os pontos como arrays [x, y]. O leitor ignora chaves e nomes desconhecidos.

Documento

Chave	Valor
`format`	Sempre "sgcDiagram".
`version`	A versão do formato, 1. Um arquivo de uma versão mais nova gera EsgcDiagramError.
`gridSize`	GridSize (10).
`theme`	"light" (padrão), "dark" ou "custom".
`layers`	Array de { name, visible (true), locked (false) }.
`nodes`	Array de nós, em ordem z.
`connectors`	Array de conectores, em ordem z.
`images`	Array de { key, png }, os dados PNG em Base64.

NodeStyle, ConnectorStyle, DefaultConnectorKind, Routing e UndoLimit são propriedades do objeto de documento e não são gravadas no arquivo.

Nó

Chave	Propriedade
`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, veja abaixo
`ports`	Array of { id, x (0.5), y (0.5), side ("auto"), flow ("both"), caption }

Conector

Chave	Propriedade
`id`	Id
`kind`	Kind: "straight", "orthogonal" (padrão) ou "curved"
`source`, `target`	{ node, port }: SourceNodeId e SourcePortId, TargetNodeId e TargetPortId
`sourcePoint`, `targetPoint`	SourcePoint, TargetPoint das extremidades livres
`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, veja abaixo
`points`	A rota armazenada, mantida ao carregar

Style

Chave	Propriedade
`fill`, `stroke`	Fill, Stroke
`strokeWidth`	StrokeWidth (1)
`dash`	Dash: "solid" (padrão), "dash" ou "dot"
`radius`	CornerRadius
`text`	{ size, bold, color }: o estilo do Text

Um arquivo pequeno completo é mostrado em Serialização.

Exemplo

```
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;
```

Veja Também

- Serialização
- TsgcDiagramDocument

Layout em Camadas

TsgcDiagramLayeredLayout (unit sgcDiagram_Layout) organiza um grafo dirigido em camadas, o método de Sugiyama: fluxogramas, diagramas de processo, máquinas de estado e grafos de dependência. Os conectores vão de uma camada para a seguinte, e o layout mantém seus cruzamentos baixos.

Propriedades

Propriedade	Significado
Direction	dglTopToBottom (padrão), dglLeftToRight, dglBottomToTop ou dglRightToLeft.
LayerSpacing	O espaço entre camadas, 60 unidades de mundo por padrão, somado ao nó mais profundo da camada.
NodeSpacing	O espaço entre nós vizinhos, 30 por padrão.
SelectionOnly	Organiza somente os nós selecionados.
Crossings	Somente leitura. Os cruzamentos de conectores contados após Execute.

Como funciona

- Os ciclos são quebrados invertendo as arestas de retorno de uma busca em profundidade na ordem da coleção.
- Cada nó recebe a camada de seu caminho mais longo a partir de uma fonte; um conector longo recebe um nó fictício em cada camada que cruza.
- Varreduras por baricentro reduzem os cruzamentos e a melhor ordem é mantida.
- As coordenadas são posicionadas em direção à mediana dos vizinhos, com conectores longos retos, sem sobreposições.
- As posições fictícias viram os waypoints dos conectores retos e curvos; os conectores ortogonais são roteados pelo roteador.
- As partes não conectadas são organizadas lado a lado, separadas por NodeSpacing.

Notas

- O layout atua sobre os nós de nível superior, ou sobre os nós selecionados quando SelectionOnly é True (por exemplo, os filhos de um contêiner). Um contêiner conta como um nó de seu tamanho e carrega seus filhos.
- Os nós mantêm o canto superior esquerdo de seus limites. O resultado é aplicado em uma transação, de modo que um layout é um passo de desfazer, e os conectores movidos são roteados novamente, também quando Routing.AutoRoute é False.
- Ogni coordenada é calculada em centésimos inteiros e toda ordenação é estável, de modo que um layout dá o mesmo resultado no Win32, no Win64 e no Linux64.

Edição Community

Os layouts requerem SGC_DIAGRAM_ADVANCED: na edição Community a unit sgcDiagram_Layout está vazia.

Exemplo

```
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;
```

Veja Também

- Layout em Árvore
- TsgcDiagramDataBinding
- Edições

Layout em Árvore

TsgcDiagramTreeLayout (unit sgcDiagram_Layout) organiza hierarquias: organogramas, árvores de categorias e mapas mentais. Os pais ficam centralizados sobre seus filhos e as subárvores vizinhas são empacotadas a NodeSpacing de distância, em tempo linear.

Propriedades

Propriedade	Significado
Direction	dglTopToBottom (padrão), dglLeftToRight, dglBottomToTop ou dglRightToLeft.
LayerSpacing	O espaço entre níveis, 60 unidades de mundo por padrão, somado ao nó mais profundo do nível.
NodeSpacing	O espaço entre subárvores vizinhas, e entre folhas empilhadas, 30 por padrão.
TreeStyle	dgoStandard (padrão) ou dgoCompactLeaves.
SelectionOnly	Organiza somente os nós selecionados.

Como funciona

- As raízes são os nós sem conector de entrada, na ordem da coleção. Um conector vai do pai para o filho; um conector que não é uma aresta da árvore é desenhado, mas ignorado no posicionamento.
- Os filhos de um nó mantêm a ordem da coleção.
- dgoCompactLeaves empilha os filhos de um nó em uma coluna à direita do centro do pai quando ele tem dois ou mais filhos e todos são folhas, o organograma clássico. Os conectores retos e curvos recebem um waypoint em cotovelo sob o pai.
- Várias árvores são posicionadas lado a lado na ordem das raízes, separadas por NodeSpacing.

Notas

- O layout atua sobre os nós de nível superior, ou sobre os nós selecionados quando SelectionOnly é True (por exemplo, os filhos de um contêiner). Um contêiner conta como um nó de seu tamanho e carrega seus filhos.
- Os nós mantêm o canto superior esquerdo de seus limites. O resultado é aplicado em uma transação, de modo que um layout é um passo de desfazer, e os conectores movidos são roteados novamente, também quando Routing.AutoRoute é False.
- Ogni coordenada é calculada em centésimos inteiros e toda ordenação é estável, de modo que um layout dá o mesmo resultado no Win32, no Win64 e no Linux64.

Edição Community

Os layouts requerem SGC_DIAGRAM_ADVANCED: na edição Community a unit sgcDiagram_Layout está vazia.

Exemplo

```
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;
```

Veja Também

- Layout em Camadas
- Vínculo de Dados

Vínculo de Dados

Um TsgcDiagramDataBinding transforma as linhas de um dataset em um diagrama e o mantém atualizado. Esta página mostra, passo a passo, a configuração usual de um organograma.

Passos

1. Dê aos dados um campo de chave e um campo de pai: cada linha é um nó, e a chave do pai dá o conector do pai. As linhas sem um pai conhecido são raízes.
2. Solte um TsgcDiagramDataBinding no formulário, defina DataSource, KeyField, ParentField e LabelField, e defina Target como a view do diagrama.
3. Escolha o layout: dgbAuto escolhe o layout em árvore para uma árvore e o layout em camadas quando uma linha tem vários pais (chaves separadas por ParentSeparator). Defina Direction e TreeStyle conforme necessário.
4. Dê estilo aos nós em OnCreateNode: o DataSet está posicionado na linha do nó, de modo que qualquer campo pode definir o texto, as cores ou o Data do nó.
5. Defina Active como True. O diagrama é construído de imediato e reconstruído quando os dados mudam.

Atualizações ao vivo

- Com LiveUpdates True, abrir o DataSet, uma edição, inserção ou exclusão confirmada, uma mudança na contagem de linhas ou no filtro, ou uma notificação da tabela reconstrói o diagrama. As solicitações são agrupadas em uma reconstrução por volta do laço de mensagens.
- A navegação não reconstrói, de modo que a seleção sobrevive a ela. Um refresh que altera apenas linhas diferentes da atual, e não a contagem de linhas, não é detectado: chame Refresh depois dele.
- Uma reconstrução espera enquanto o DataSet está sendo editado e enquanto o usuário arrasta na view. Fechar o DataSet mantém o diagrama.
- No Delphi 10.2 e posterior, a reconstrução é enfileirada com TThread.ForceQueue: aplicativos de console e serviços devem chamar CheckSynchronize.
- Uma reconstrução limpa o histórico de desfazer: as alterações feitas no diagrama não são gravadas de volta nos dados.

Edição Community

O vínculo de dados requer SGC_DIAGRAM_ADVANCED. Na edição Community, construa o diagrama no código com AddNode e AddConnector.

Exemplo

```
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;
```

Veja Também

- TsgcDiagramDataBinding
- Layout em Árvore
- Layout em Camadas

Formas

Todo nó tem uma Shape do tipo TsgcDiagramShape. A forma decide o contorno, a área que recebe o texto, as portas padrão e as cores do tema. Um nó é posicionado com Left, Top, Width e Height em unidades de mundo e girado com Angle, em graus no sentido horário em torno de seu centro.

Forma	Uso
dgsRectangle, dgsRoundRect, dgsEllipse	Caixas básicas. dgsRoundRect usa um raio de canto de 8 quando Style.CornerRadius é 0; as outras caixas só arredondam os cantos quando CornerRadius é maior que 0.
dgsProcess, dgsDecision, dgsTerminator, dgsData, dgsDocument, dgsPredefinedProcess, dgsConnectorCircle	Os símbolos de fluxograma. Um terminador é sempre arredondado pela metade de seu lado menor; uma decisão é um losango cujas portas padrão são seus cantos.
dgsText	Texto livre sem preenchimento nem borda.
dgsImage	Uma imagem do documento, nomeada por ImageKey. Veja abaixo.
dgsPath	Um contorno personalizado a partir de PathData, na sintaxe de caminho SVG dentro de uma caixa de 0 a 100 que é esticada até o nó.
dgsContainer	Uma caixa com uma faixa de cabeçalho que guarda outros nós. Um nó pertence a ela por ContainerId, e mover o contêiner move seus filhos.

Texto

O texto é desenhado dentro do retângulo de texto da forma, com quebra de linha, com TextAlign (thaCenter por padrão) e TextVAlign (tvaMiddle por padrão). O tamanho, o peso e a cor da fonte vêm de Style.Text. Na view, um clique duplo, F2 ou um caractere digitado edita o texto no local.

Imagens

Images.AddPNG(aKey, aStream) guarda os bytes PNG de um stream e os decodifica; Images.Add(aKey, aImage) adiciona um TsgcUIImage decodificado. Um nó com Shape dgsImage e ImageKey definido como a chave desenha a imagem esticada até sua caixa. Outros formatos são decodificados pelo seu manipulador do evento OnDecodeImage do documento. Excluir uma imagem que um nó ainda usa gera EsgcDiagramError.

Grupos e contêineres

Document.Group coloca os nós selecionados em um grupo e retorna seu GroupId; um clique então seleciona o grupo inteiro, e Alt com um clique seleciona um membro. Ungroup remove o grupo da seleção. Um contêiner possui seus filhos: eles se movem com ele, um contêiner excluído os exclui, e um nó solto sobre um contêiner na view torna-se seu filho.

Exemplo

```
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;
```

Veja Também

- Portas e Fixação
- Temas
- Conectores e Roteamento
- TsgcDiagramNode

Conectores e Roteamento

Um TsgcDiagramConnector une uma origem e um destino. Cada extremidade está fixada a um nó, por SourceNodeId e TargetNodeId, ou livre em SourcePoint e TargetPoint. Document.AddConnector(aSource, aTarget) cria um conector do tipo DefaultConnectorKind (dgcOrthogonal por padrão) e o roteia no fim do conjunto de alterações.

Tipo	Rota
dgcStraight	Uma linha reta, ou uma polilinha pelos Waypoints.
dgcOrthogonal	Segmentos horizontais e verticais. Na edição Registered, o roteador evita os nós com uma margem e afasta os segmentos paralelos; na Community, a rota é um cotovelo simples.
dgcCurved	Uma curva suave, pelos Waypoints quando há algum.

Setas, linha e rótulos

- StartArrow (dgaNone por padrão) e EndArrow (dgaTriangle por padrão) aceitam dgaNone, dgaTriangle, dgaOpenTriangle, dgaDiamond, dgaOpenDiamond ou dgaCircle. ArrowSize define seu tamanho.
- Style.Stroke, Style.StrokeWidth e Style.Dash (dgdSolid, dgdDash, dgdDot) definem a linha.
- Labels guarda qualquer número de TsgcDiagramLabel. Position vai de 0 na origem a 1 no destino, Offset afasta o rótulo da linha. O usuário arrasta um rótulo ao longo de seu conector.

Roteamento

O documento roteia todo conector cujos nós, extremidades, tipo ou waypoints mudaram quando o EndUpdate mais externo é executado, enquanto Routing.AutoRoute é True. RouteAll roteia todos os conectores novamente. A rota está em Points; RouteFailed é True quando o roteador ortogonal não encontrou caminho ao redor dos nós e desenhou uma alternativa.

Propriedade de roteamento	Padrão	Significado
Margin	12	A distância que uma rota ortogonal mantém dos nós.
BendPenalty	40	O custo de uma curva, de modo que o roteador prefere menos curvas a um caminho um pouco mais curto.
Spacing	6	A distância entre segmentos paralelos que compartilham um corredor.
AutoRoute	True	Roteia no fim de cada conjunto de alterações. Desative enquanto constrói um diagrama grande no código e depois chame RouteAll.

Exemplo

```
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;
```

Veja Também

- Portas e Fixação
- Edições
- Formas
- TsgcDiagramConnector

Portas e Fixação

Uma porta é um ponto em um nó onde a extremidade de um conector se prende. Um nó com uma coleção Ports vazia usa as portas padrão de sua forma: `left`, `top`, `right` e `bottom`, os pontos médios dos lados (os cantos de uma decisão). Adicione suas próprias portas para dar a um nó entradas e saídas tipadas.

TsgcDiagramPort	Significado
Id	Nome da porta, único em seu nó. Um conector se refere a ela em SourcePortId ou TargetPortId.
X, Y	Posição como fração da caixa do nó, de 0 a 1.
Side	dgpAuto, dgpLeft, dgpTop, dgpRight ou dgpBottom: a direção em que uma rota sai da porta.
Flow	dgfBoth, dgfIn ou dgfOut.
Caption	Texto desenhado dentro do nó ao lado da porta. As legendas ficam ocultas abaixo de zoom 0.4.

Fixação

- Uma extremidade de conector com um Id de porta é fixada a essa porta.
- Uma extremidade de conector com um Id de porta vazio é fixada ao contorno do nó: a extremidade acompanha o nó e encontra o contorno na linha em direção ao centro.
- Uma extremidade de conector sem nó é livre em SourcePoint ou TargetPoint.
- Node.PortPoint(aPortId) retorna a posição de mundo de uma porta.

Na view

ShowPorts decide quando as portas são desenhadas: dgvNever, dgvHover (o padrão, as portas do nó sob o ponteiro) ou dgvAlways. Um arraste a partir de uma porta desenha um novo conector e, enquanto ele é desenhado, toda porta é um destino. Em um nó selecionado, as alças de redimensionamento cobrem as portas laterais, portanto defina Tool como dgtConnector para desenhar a partir dele. Antes de uma extremidade ser fixada, a view dispara OnConnecting, onde você pode recusá-la; depois que o conector é criado, dispara OnConnected.

Exemplo

```
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;
```

Veja Também

- Conectores e Roteamento
- Edição e Teclado
- TsgcDiagramPort

Edição e Teclado

A view edita o documento por meio de seu Controller. Toda ação do usuário que altera o documento é um passo de desfazer, e Esc durante um arraste o cancela.

Ferramentas

Ferramenta	Efeito
dgtSelect	O padrão. Clique para selecionar, arraste para mover, as alças redimensionam e giram, um arraste em espaço vazio desenha um retângulo de seleção, um arraste a partir de uma porta desenha um conector.
dgtPan	Um arraste com o botão esquerdo faz a rolagem.
dgtConnector	Um arraste a partir de qualquer nó desenha um conector.
dgtShape	Um clique ou um arraste cria um nó da forma em ShapeTool. A ferramenta continua ativa depois de cada nó; Esc volta para dgtSelect. Um clique em um item da paleta define esta ferramenta.

Mouse

- Shift e um clique adicionam à seleção, Ctrl e um clique alternam um item, Alt e um clique selecionam um membro de um grupo.
- Shift ao redimensionar mantém a proporção; Shift ao girar ajusta o ângulo em passos de 15 graus.
- A roda rola na vertical, Shift e a roda na horizontal, Ctrl e a roda fazem zoom em torno do ponteiro.
- O botão do meio, ou o botão esquerdo com Espaço pressionado, faz a rolagem.
- O botão direito seleciona o item sob o ponteiro, pronto para um menu popup.
- Um clique duplo em um nó edita seu texto e dispara OnItemDblClick.

Teclado

Teclas	Ação
Ctrl+Plus, Ctrl+Minus	Zoom para mais e para menos em torno do centro da view.
Ctrl+0, Ctrl+1	Zoom 1, zoom para ajustar.
Ctrl+A, Ctrl+C	Selecionar tudo, copiar.
Ctrl+X, Ctrl+V, Ctrl+D	Recortar, colar, duplicar.
Ctrl+Z, Ctrl+Y ou Ctrl+Shift+Z	Desfazer, refazer.
Ctrl+G, Ctrl+Shift+G	Agrupar, desagrupar.
Delete	Exclui a seleção (os itens bloqueados permanecem).
Setas	Move a seleção por GridSize; com Alt, por uma unidade de mundo.

F2, ou qualquer caractere	Edita o texto do nó selecionado.
Tab, Shift+Tab	Seleciona o próximo item ou o item anterior.
Esc	Cancela um arraste, sai da ferramenta atual ou limpa a seleção.

No modo ReadOnly, só funcionam as teclas de zoom, Ctrl+A, Ctrl+C, Tab e Esc.

Edição de texto

O texto é editado em um memo sobre o nó. Enter ou um clique em outro lugar confirma, Esc cancela, Shift+Enter inicia uma nova linha. OnTextEditing pode recusar a edição definindo aAllow como False, e OnTextEdited dispara após uma confirmação. EditText(aNode) inicia o editor a partir do código.

Exemplo

```
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;
```

Veja Também

- Ajuste e Guias
- Desfazer e Área de Transferência
- Modo Visualizador
- TsgcDiagramController
- Eventos

Ajuste e Guias

Dois recursos alinham os nós enquanto o usuário os move, redimensiona ou cria.

Propriedade	Efeito
Diagram.GridSize	Espaçamento da grade em unidades de mundo, 10 por padrão. As setas movem a seleção por essa quantidade.
ShowGrid	Desenha a grade na view.
SnapToGrid	Ajusta à grade os nós movidos, redimensionados e criados, e as formas soltas.
SnapToGuides	Mostra guias de alinhamento: enquanto um nó se move ou é redimensionado, suas bordas e linhas centrais se ajustam às bordas e linhas centrais dos nós próximos e uma linha de guia é desenhada.

- Uma guia captura uma borda ou um centro a até 5 pixels do dispositivo, em relação aos nós a até 400 pixels do dispositivo.
- As guias se aplicam a nós que não estão girados.
- Shift ao redimensionar desativa as guias nesse arraste.
- Alt com uma tecla de seta move a seleção por uma unidade de mundo, fora da grade.

Exemplo

```
uses
  sgcDiagram_VCL;

procedure SetupSnapping(aView: TsgcVCLDiagramView);
begin
  aView.Diagram.GridSize := 20;
  aView.ShowGrid := True;
  aView.SnapToGrid := True;
  aView.SnapToGuides := True;
end;
```

Veja Também

- Edição e Teclado
- Formas

Desfazer e Área de Transferência

O documento mantém uma pilha de desfazer e uma pilha de refazer. Cada ação do usuário na view é um passo. No código, BeginTransaction(aCaption) e EndTransaction agrupam qualquer número de alterações em um passo, e Rollback descarta as alterações da transação aberta.

- Undo, Redo, CanUndo, CanRedo, UndoCaption e RedoCaption atuam sobre as pilhas; a view e o controlador também têm Undo e Redo.
- UndoLimit (100 por padrão) é o número de passos mantidos.
- Uma alteração feita pelo código fora de uma transação limpa as duas pilhas, de modo que um desfazer nunca restaura um estado obsoleto.
- BeginUndoSuspend e EndUndoSuspend envolvem uma reconstrução em massa que não deve ser registrada: eles limpam as pilhas e não geram passo, e as views continuam sendo notificadas. Eles podem ser aninhados.
- Document.Clear não é um passo de desfazer.

Área de transferência

A área de transferência é interna ao aplicativo. CopySelection retorna os nós selecionados, os filhos dos contêineres selecionados, os conectores cujas duas extremidades são copiadas e as imagens que eles usam, como um fragmento JSON, e o guarda na área de transferência interna. Paste(aJSON) insere um fragmento como um único passo de desfazer e seleciona os itens colados; Paste sem argumento cola a área de transferência interna, 20 unidades de mundo mais adiante a cada colagem repetida. A view tem CopyToClipboard, CutToClipboard, PasteFromClipboard e DeleteSelection, e Ctrl+D duplica a seleção.

Exemplo

```
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;
```

Veja Também

- Serialização
- Edição e Teclado

Camadas

As camadas organizam os itens de um diagrama, por exemplo uma camada de fundo com molduras e notas sob a camada de trabalho. Cada nó e conector indica sua camada na propriedade Layer; um nome vazio é a camada padrão.

TsgcDiagramLayer	Significado
Name	Nome usado pela propriedade Layer dos itens.
Visible	False oculta os itens da camada na view e na exportação.
Locked	True impede que os itens da camada sejam selecionados e editados.

- Layers.Add, Layers.Find(aName) e os Items da coleção gerenciam as camadas. Adicionar, excluir e renomear uma camada são passos de desfazer; Visible e Locked não são.
- Document.LayerVisible(aName) e LayerLocked(aName) indicam como uma camada é tratada.
- Um único item também pode ser bloqueado com sua própria propriedade Locked.

Edição Community

Sem SGC_DIAGRAM_ADVANCED só uma camada pode ser adicionada: um segundo Layers.Add gera EsgcDiagramError. Um arquivo com mais camadas ainda é carregado e salvo sem alteração, mas toda camada é desenhada, exportada e editada como visível e desbloqueada.

Exemplo

```
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;
```

Veja Também

- Edições
- Modo Visualizador
- TsgcDiagramLayer

Serialização

Um documento é salvo e carregado como JSON. O formato é o mesmo em todas as plataformas e em todas as edições.

Método	Efeito
SaveToJSON, LoadFromJSON	Para e a partir de uma string.
SaveToFile, LoadFromFile	Para e a partir de um arquivo.
SaveToStream, LoadFromStream	Para e a partir de um stream.
CopySelection, Paste	Um fragmento do documento, veja Desfazer e Área de Transferência.

- A saída é canônica: uma ordem de chaves fixa, propriedades no valor padrão omitidas, sem indentação. O mesmo documento sempre gera o mesmo texto.
- As imagens do documento são guardadas como dados PNG no arquivo.
- O leitor é tolerante: chaves desconhecidas e nomes de enumeração desconhecidos são ignorados. Um arquivo gravado por uma versão de formato mais nova gera EsgcDiagramError, e um erro de análise informa sua posição.
- Carregar substitui o conteúdo do documento. As rotas dos conectores guardadas no arquivo são mantidas.
- Uma view guarda seu diagrama no arquivo dfm ou fmx como a propriedade DiagramJSON, somente quando o diagrama não está vazio.

Exemplo

```
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;
```

Um arquivo pequeno

Dois nós e um conector, com indentação aqui para facilitar a leitura:

```
{
  "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]]
    }
  ]
}
```

Veja Também

- Desfazer e Área de Transferência
- Exportação
- Formato JSON

Exportação

Um diagrama é exportado para PNG e SVG. A imagem cobre os limites do conteúdo do documento mais uma margem.

Método	Notas
View.SaveToPNG(aFileName or aStream, aScale)	Exporta o diagrama da view com seu FontFile e OnDrawNode aplicados. aScale é o número de pixels por unidade de mundo, 1 por padrão.
View.SaveToSVG(aFileName or aStream, aScale)	O mesmo como SVG.
TsgcDiagramRenderer.SaveToPNGFile, SaveToPNGStream	(aDocument, aFileName ou aStream, aScale = 1, aMargin = 20). Não precisa de view.
TsgcDiagramRenderer.SaveToSVGFile, SaveToSVGStream	O mesmo como SVG.
TsgcDiagramRenderer.RenderTo	(aCanvas, aDocument, aViewport) pinta uma parte do documento em um TsgcUICanvas do sgcUICore.

- A exportação usa o tema do documento, de modo que um diagrama escuro é exportado escuro.
- O renderizador funciona em um aplicativo de console, em um serviço e no Linux64; veja Início Rápido Sem Janela.
- A exportação para PDF e a impressão não fazem parte da versão 1.

Exemplo

```
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;
```

Veja Também

- Temas
- Serialização
- TsgcDiagramRenderer

Temas

As views, as paletas e a exportação são desenhadas com os temas do sgcUICore, de modo que um diagrama combina com os outros produtos sgcUI no mesmo formulário.

- **Theme** da view, da paleta e do documento é utLight (padrão), utDark ou utCustom. O Theme de uma view define o Theme de seu Diagram.
- **Style** de cada nó e conector: Fill, Stroke, StrokeWidth, Dash, CornerRadius e Text (Size, Bold e Color). Uma cor é um TsgcUIColor ARGB, por exemplo sgcUIARGB($FF, $00, $96, $88); 0 mantém a cor do tema, que depende da forma.
- **NodeStyle** e **ConnectorStyle** do documento são os modelos copiados para cada novo nó e conector.
- **FontFile** da view desenha o texto com um arquivo TrueType em vez da fonte embutida no sgcUICore.
- **OnDrawNode** da view permite pintar um nó você mesmo, ou acrescentar ao desenho padrão: defina aDefault como False para pulá-lo.

Exemplo

```
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;
```

Veja Também

- Exportação
- Formas
- TsgcDiagramStyle

Modo Visualizador

Defina ReadOnly como True para mostrar um diagrama que o usuário pode explorar mas não alterar, como um mapa de processo em um painel ou um organograma gerado.

- O usuário pode selecionar itens, fazer zoom com Ctrl e a roda ou as teclas, rolar com a roda, o botão do meio, Espaço ou a ferramenta de rolagem, e copiar a seleção.
- Nada é movido, redimensionado, conectado, excluído, colado ou editado, e a paleta não pode soltar formas.
- O ponteiro vira uma mão sobre um item, e OnItemClick, OnItemDblClick e OnHover informam o item, de modo que o aplicativo pode abrir uma página de detalhes ou mostrar uma dica.
- ZoomToFit, ZoomToSelection e CenterOn movem a view a partir do código; MinZoom (0.05) e MaxZoom (8) limitam o zoom.

Para proteger apenas alguns itens, deixe ReadOnly como False e defina Locked nesses itens ou em sua camada.

Exemplo

```
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;
```

Veja Também

- Edição e Teclado
- Camadas
- TsgcVCLDiagramView
- TsgcFMXDiagramView

Visão geral do .NET

sgcDiagram .NET é uma portagem C# gerenciada da biblioteca sgcDiagram do Delphi, não um wrapper em torno dela. As classes C# vivem no namespace **esegece.sgcDiagram** e mantêm os nomes do Delphi, de modo que **TsgcDiagramDocument**, **TsgcDiagramNode**, **TsgcDiagramConnector**, **TsgcDiagramController**, **TsgcDiagramRenderer** e as enumerações se leem igual nas duas linguagens. A view e a paleta seguem o padrão de nomes do Delphi em cada host: **TsgcVCLDiagramView** é **TsgcWinFormsDiagramView**, **TsgcWpfDiagramView** e **TsgcAvaloniaDiagramView**, e **TsgcVCLDiagramPalette** é **TsgcWinFormsDiagramPalette**, **TsgcWpfDiagramPalette** e **TsgcAvaloniaDiagramPalette**. Os controles são desenhados com a portagem .NET do **sgcUICore**, o mesmo canvas, as mesmas fontes e os mesmos temas claro e escuro dos outros produtos da suíte visual da eSeGeCe.
O documento, a pilha de desfazer, o formato JSON, o roteador, os layouts, o pintor e o controller são o código Delphi portado linha a linha. Um diagrama salvo por uma aplicação .NET abre em uma aplicação Delphi e vice-versa, o texto JSON e a exportação SVG são os mesmos do Delphi, e os layouts e as rotas dão as mesmas coordenadas.

Pacotes

A versão é dividida em cinco pacotes NuGet, de modo que um projeto só traz o host que usa.

PackageId	Contém	Frameworks de destino
esegece.sgcDiagram	O documento com seus nós, conectores, portas, rótulos, camadas, imagens e estilos, desfazer e refazer, o formato JSON e a área de transferência interna, o roteador, os layouts em camadas e em árvore, o modelo da paleta, o controller, o pintor, a exportação SVG do **TsgcDiagramRenderer** e o **TsgcDiagramDataBinding**. Sem dependência nativa. Depende de esegece.sgcUICore.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDiagram.Rendering	A saída PNG: os métodos de extensão **SaveToPNGStream** e **SaveToPNGFile** do **TsgcDiagramRenderer** e o raster do cache da view, desenhados no canvas SkiaSharp de esegece.sgcUICore.Rendering, no Windows, Linux e macOS.	netstandard2.0, net472, net6.0, net8.0, net9.0
esegece.sgcDiagram.WinForms	**TsgcWinFormsDiagramView** e **TsgcWinFormsDiagramPalette**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDiagram.Wpf	**TsgcWpfDiagramView** e **TsgcWpfDiagramPalette**.	net472, net8.0-windows, net9.0-windows
esegece.sgcDiagram.Avalonia	**TsgcAvaloniaDiagramView** e **TsgcAvaloniaDiagramPalette**, no Windows, Linux e macOS com Avalonia 11.	net8.0, net9.0

Os três pacotes de host dependem de esegece.sgcDiagram.Rendering, que depende de esegece.sgcDiagram, então uma linha `PackageReference` traz tudo o que um host precisa. Veja Instalação pelo NuGet.

Edições

Edição	Ids dos pacotes	Inclui
Registered	esegece.sgcDiagram, .Rendering, .WinForms, .Wpf e .Avalonia	Tudo: formas, conectores, portas e fixação, edição, encaixe e guias, desfazer e área de transferência, camadas, JSON, exportação PNG e SVG, temas, o modo visualizador e os recursos avançados: os layouts em camadas e em árvore, o roteamento ortogonal com desvio de obstáculos e ajuste, o vínculo de dados e mais de uma camada.
Community	Os mesmos ids com o sufixo .Community, por exemplo esegece.sgcDiagram.WinForms.Community.	Tudo exceto os recursos avançados. **TsgcDiagramLayeredLayout**, **TsgcDiagramTreeLayout** e **TsgcDiagramDataBinding** não estão nos pacotes Community, os conectores **dgcOrthogonal** são roteados como cotovelos simples, e **Layers.Add** gera **EsgcDiagramError** quando uma camada já existe.

É a mesma divisão das edições do Delphi, e um arquivo de diagrama é o mesmo em ambas, veja Edições.

Hosts

Host	Equivalente no Delphi	Plataformas
WinForms	VCL (TsgcVCLDiagramView, TsgcVCLDiagramPalette)	Windows, .NET Framework 4.7.2 e .NET 8 e 9.
WPF	VCL e FMX	Windows, .NET Framework 4.7.2 e .NET 8 e 9. Os controles podem ser declarados em XAML.
Avalonia	FMX (TsgcFMXDiagramView, TsgcFMXDiagramPalette), com zoom por pinça e deslocamento com um dedo em tela de toque	Windows, Linux e macOS, .NET 8 e 9.
Sem janela	TsgcDiagramDocument and TsgcDiagramRenderer	Qualquer plataforma dos frameworks de destino: JSON e SVG com esegece.sgcDiagram, PNG com esegece.sgcDiagram.Rendering.

Todo host tem a mesma API: as propriedades, métodos e eventos da view e da paleta do Delphi com os mesmos nomes. Veja Classes do .NET para os nomes das classes e Diferenças do .NET em relação ao Delphi para o que muda em relação ao Delphi.

Veja também

- Instalação pelo NuGet
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Vinculação de dados no .NET

- Renderização sem janela no .NET
- Demos do .NET
- Plataformas

Instalação pelo NuGet

Os pacotes estão no nuget.org. Adicione o pacote do host, os pacotes do núcleo e da renderização vêm com ele como dependências.

Edição Registered

```
dotnet add package esegece.sgcDiagram.WinForms
dotnet add package esegece.sgcDiagram.Wpf
dotnet add package esegece.sgcDiagram.Avalonia
```

Edição Community

```
dotnet add package esegece.sgcDiagram.WinForms.Community
dotnet add package esegece.sgcDiagram.Wpf.Community
dotnet add package esegece.sgcDiagram.Avalonia.Community
```

Ou adicione a referência ao arquivo de projeto manualmente:

```
<ItemGroup>
  <PackageReference Include="esegece.sgcDiagram.Wpf" Version="2026.10.0" />
</ItemGroup>
```

Referencie os pacotes Registered ou os pacotes Community em um projeto, nunca os dois: eles contêm os mesmos tipos no mesmo namespace.

Configurações do projeto

- WinForms: um projeto `net8.0-windows` com `<UseWindowsForms>true</UseWindowsForms>`, ou um projeto .NET Framework 4.7.2.
- WPF: um projeto `net8.0-windows` com `<UseWPF>true</UseWPF>`, ou um projeto .NET Framework 4.7.2. Mapeie o namespace no XAML com `xmlns:sgc="clr-namespace:esegece.sgcDiagram;assembly=esegece.sgcDiagram.Wpf"`.
- Avalonia: um projeto `net8.0` ou `net9.0` com os pacotes **Avalonia.Desktop** e **Avalonia.Themes.Fluent** do Avalonia 11.3, como qualquer aplicação desktop Avalonia. O editor de texto no local é um **TextBox** do Avalonia e toma o seu visual do tema Fluent.
- Sem janela: esegece.sgcDiagram sozinho basta para o documento, o JSON e o SVG. Adicione esegece.sgcDiagram.Rendering para PNG, que traz o SkiaSharp e seus recursos nativos para Windows e Linux.

Em C# todo host precisa de `using esegece.sgcDiagram;`, e de `using esegece.sgcUICore;` para **TsgcUITheme**, **TsgcUINotifyEvent** e os outros tipos do sgcUICore.

Veja também

- Visão geral do .NET
- Início rápido WinForms
- Início rápido WPF
- Início rápido Avalonia
- Demos do .NET

Início rápido WinForms

Este início rápido constrói um formulário em código com uma paleta de formas à esquerda, uma view de diagrama ocupando o resto e um pequeno fluxograma na view. Arraste uma forma da paleta para a view, ou clique nela e depois na view, e conecte dois nós a partir de uma porta.

Passos

1. Crie um projeto Windows Forms para `net8.0-windows` e adicione o pacote **esegece.sgcDiagram.WinForms** (**esegece.sgcDiagram.WinForms.Community** na edição Community).
2. Adicione `using esegece.sgcDiagram;` e `using esegece.sgcUICore;`.
3. Crie um **TsgcWinFormsDiagramView** e uma **TsgcWinFormsDiagramPalette** e adicione-os ao formulário. São controles WinForms comuns: Dock, Anchor, TabIndex e Enabled funcionam como de costume.
4. Defina **View** da paleta como a view. A paleta carrega as formas Flowchart e Basic quando é criada.
5. Preencha **Diagram**, o **TsgcDiagramDocument** da view, dentro de uma transação para que toda a configuração seja um passo de desfazer, e atribua os eventos.

Exemplo

Notas

- **Tool** escolhe o que um clique faz (**dgtSelect**, **dgtPan**, **dgtConnector**, **dgtShape** com **ShapeTool**), e **ReadOnly** transforma a view em um visualizador, como no Delphi.
- O teclado é o do Delphi, veja Edição e Teclado. O editor no local é uma **TextBox** de várias linhas, exposta como **TextEditor**: Enter confirma, Shift+Enter é uma nova linha, Esc cancela.
- **Theme** é por controle, então defina-o na view e na paleta. A view o repassa ao seu documento.
- O diagrama é armazenado na propriedade oculta **DiagramJSON**, de modo que uma view colocada com o designer mantém o diagrama desenhado nela.
- **SaveToPNG** e **SaveToSVG** gravam o diagrama inteiro em um arquivo ou stream, em uma escala.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido VCL

Início rápido WPF

Os controles WPF podem ser declarados em XAML. Esta janela tem a paleta à esquerda e a view ocupando o resto; o code behind liga a paleta à view e adiciona um pequeno fluxograma.

Passos

1. Crie um projeto WPF para `net8.0-windows` e adicione o pacote **esegece.sgcDiagram.Wpf** (**esegece.sgcDiagram.Wpf.Community** na edição Community).
2. Mapeie o namespace na janela: `xmlns:sgc="clr-namespace:esegece.sgcDiagram;assembly=esegece.sgcDiagram.Wpf"`.
3. Declare uma **TsgcWpfDiagramPalette** e uma **TsgcWpfDiagramView** e defina as suas propriedades como atributos.
4. No code behind defina **View** da paleta, preencha **Diagram** e atribua os 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

- As propriedades são propriedades CLR simples, então o XAML as define como valores de atributo. **View** da paleta recebe o objeto view, defina-o em código.
- Um diagrama pode ser dado em XAML com o atributo **DiagramJSON**. O texto JSON começa com uma chave, então escreva-o depois do escape `{}`.
- As coordenadas da API, inclusive **ScreenToDiagram** e **DiagramToScreen**, são pixels de dispositivo do controle, isto é, pixels independentes de dispositivo vezes a escala de DPI.
- Chame **Dispose** da view quando a janela fechar: ele para o timer da view e libera o seu controller.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido VCL

Início rápido Avalonia

Os controles Avalonia rodam no Windows, Linux e macOS. Este início rápido constrói uma janela em código com a paleta à esquerda e a view ocupando o resto, no tema escuro.

Passos

1. Crie um projeto desktop Avalonia para `net8.0` com o Avalonia 11.3 e o tema Fluent, e adicione o pacote **esegece.sgcDiagram.Avalonia** (**esegece.sgcDiagram.Avalonia.Community** na edição Community).
2. Crie um **TsgcAvaloniaDiagramView** e uma **TsgcAvaloniaDiagramPalette**, defina **View** da paleta e coloque ambos em um painel.
3. Preencha **Diagram** e atribua os eventos.

Exemplo

Notas

- Os eventos dos controles Avalonia são propriedades delegate, atribuídas com `=`, enquanto os controles WinForms e WPF têm eventos C#, atribuídos com `+=`.
- **Theme** dos controles Avalonia oculta a propriedade **Theme** do Avalonia (o tema do controle) e recebe um **TsgcUITheme**.
- Em uma tela de toque, um gesto de pinça dá zoom e um dedo no espaço vazio desloca; um dedo em um nó o arrasta, como na view FMX.
- Um arrasto da paleta usa `DragDrop.DoDragDropAsync` com o formato de aplicação **DragFormat** da view (**CS_DIAGRAM_AVALONIA_DRAG_FORMAT**). **DragPayloadOf** e **CreateDragData** leem e constroem esses dados para uma origem de arrasto própria.
- O Avalonia não descarta controles, então chame **Dispose** da view quando a janela fechar.

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Classes do .NET
- Diferenças do .NET em relação ao Delphi
- Início Rápido FMX

Classes do .NET

As classes C# têm os nomes das classes do Delphi. As classes do modelo, dos motores e do pintor são as mesmas em todo host; os controles têm uma classe por host.

Controles

Delphi	WinForms	WPF	Avalonia
TsgcVCLDiagramView, TsgcFMXDiagramView	TsgcWinFormsDiagramView	TsgcWpfDiagramView	TsgcAvaloniaDiagramView
TsgcVCLDiagramPalette, TsgcFMXDiagramPalette	TsgcWinFormsDiagramPalette	TsgcWpfDiagramPalette	TsgcAvaloniaDiagramPalette
Classe base	TsgcUIWinFormsControl	TsgcUIWpfControl	TsgcUIAvaloniaControl

A view tem as propriedades do Delphi **Diagram**, **Controller**, **Tool**, **ShapeTool**, **ReadOnly**, **Zoom**, **MinZoom**, **MaxZoom**, **ShowGrid**, **SnapToGrid**, **SnapToGuides**, **ShowPorts**, **Theme** e **FontFile**, além de **DiagramJSON** e **TextEditor**; os métodos **ZoomToFit**, **ZoomToSelection**, **CenterOn**, **ScreenToDiagram**, **DiagramToScreen**, **Undo**, **Redo**, **CopyToClipboard**, **CutToClipboard**, **PasteFromClipboard**, **DeleteSelection**, **EditText**, **SaveToPNG** e **SaveToSVG**; e os eventos **OnChange**, **OnSelectionChange**, **OnItemClick**, **OnItemDblClick**, **OnHover**, **OnConnecting**, **OnConnected**, **OnDrawNode**, **OnTextEditing**, **OnTextEdited** e **OnZoomChange**. A paleta tem **Model**, **View**, **Items**, **IconSize**, **ShowCaptions**, **Theme**, **DragPayload**, **LoadDefaultItems** e **OnItemClick**.

Modelo e motores

No pacote esegece.sgcDiagram:

Classe	Descrição
TsgcDiagramDocument	O diagrama: **Nodes**, **Connectors**, **Layers**, **Images**, **Selection**, **NodeStyle**, **ConnectorStyle**, **Routing**, **Theme**, **GridSize**; **AddNode**, **AddConnector**, **BeginTransaction**, **EndTransaction**, **Undo**, **Redo**, **RouteAll**, **SaveToJSON**, **LoadFromJSON**, **SaveToFile**, **LoadFromFile**, **CopySelection**, **Paste**. Uma view cria o seu próprio, um programa sem janela cria um com `new TsgcDiagramDocument()`.
TsgcDiagramNode, TsgcDiagramConnector	Os itens, com **Style**, **Layer** e **Data**; um nó tem **Ports**, um conector tem **Labels**, como no Delphi.
TsgcDiagramPort, TsgcDiagramLabel, TsgcDiagramLayer, TsgcDiagramImage, TsgcDiagramStyle	As partes dos itens e do documento. As coleções são **TsgcDiagramNodes**, **TsgcDiagramConnectors**, **TsgcDiagramPorts**, **TsgcDiagramLabels** e **TsgcDiagramLayers**, com `Add()`, `Count` e um indexador.
TsgcDiagramUndoStack	A pilha de desfazer de um documento, acessada por **UndoStack**.
TsgcDiagramController	A lógica de edição compartilhada pelos hosts: teste de acerto, os estados de arrasto, o mapa de teclas, o encaixe, as guias e os drops. Uma view a expõe como **Controller**.

TsgcDiagramPaletteModel, TsgcDiagramPaletteItem	A paleta sem host: itens, categorias, layout, pintura, teste de acerto e o payload de arrasto.
TsgcDiagramPainter, TsgcDiagramViewCache	O pintor do documento e o raster em cache de uma view.
TsgcDiagramRenderer	Exportação sem janela. **SaveToSVGStream** e **SaveToSVGFile** estão em esegece.sgcDiagram, **SaveToPNGStream** e **SaveToPNGFile** são métodos de extensão em esegece.sgcDiagram.Rendering.
TsgcDiagramLayeredLayout, TsgcDiagramTreeLayout	Os layouts automáticos. Edição Registered.
TsgcDiagramDataBinding	Constrói um diagrama a partir de um `DataTable`, veja Vinculação de dados no .NET. Edição Registered.
EsgcDiagramError	A exceção da biblioteca.

Classes estáticas

As constantes e funções no nível de unit do Delphi são membros estáticos de uma classe com o nome da unit: **sgcDiagram_JSON** (**sgcDiagramSaveJSON**, **sgcDiagramLoadJSON**, **sgcDiagramCopy**, **sgcDiagramPaste**), **sgcDiagram_Render**, **sgcDiagram_Controller** (as constantes **CS_DIAGRAM_*** e os códigos de tecla **CS_DIAGRAM_KEY_***), **sgcDiagram_Palette** e **sgcDiagram_Types**. **sgcDiagram_Render_Rendering.RegisterRaster** instala o raster Skia; os controles o chamam em seu construtor. O formato de arrasto da paleta é **sgcDiagram_WinForms.CS_DIAGRAM_DRAG_FORMAT**, **sgcDiagram_Wpf.CS_DIAGRAM_DRAG_FORMAT** e **TsgcAvaloniaDiagramView.CS_DIAGRAM_AVALONIA_DRAG_FORMAT**, todos `esegece.sgcDiagram.Shape`.

Enumerações

As enumerações e seus valores mantêm os nomes do Delphi: **TsgcDiagramShape** (**dgsRectangle**, **dgsDecision** e as outras formas), **TsgcDiagramConnectorKind**, **TsgcDiagramArrow**, **TsgcDiagramTool**, **TsgcDiagramShowPorts**, **TsgcDiagramDirection**, **TsgcDiagramTreeStyle**, **TsgcDiagramBindingLayout**. Um valor é escrito com o seu tipo, por exemplo `TsgcDiagramShape.dgsProcess`. Os conjuntos do Delphi são enumerações `[Flags]`: **TsgcDiagramChanges** e **TsgcDiagramShiftState**.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcDiagramDocument
- TsgcDiagramController
- TsgcVCLDiagramView

Diferenças do .NET em relação ao Delphi

A API C# segue a API Delphi nome por nome. O que muda vem da linguagem e dos frameworks:

Delphi	.NET
Os eventos são ponteiros de método, Sender é o componente.	Os controles WinForms e WPF têm eventos C# (`+=`), Sender é o controle. Os controles Avalonia, o documento, o controller, o modelo da paleta e o vínculo de dados têm propriedades delegate (`=`).
Parâmetros `var` dos eventos (**OnTextEditing**, **OnConnecting**, **OnDrawNode**).	Parâmetros `ref` . Uma lambda com um parâmetro `ref` nomeia os tipos de todos os seus parâmetros.
Propriedades indexadas e valores lidos por uma função.	Métodos: **ContentBounds()**, **UndoCaption()**, **CanUndo()**, **Viewport()** do controller, **GetCollapsed**, **SetCollapsed**, **GetCategory**, **MaxScroll()** e **Payload(i)** do modelo da paleta. Um item de coleção é adicionado com `Add()` .
Argumentos **TList**.	`List<TsgcDiagramItem>` , `List<TsgcDiagramNode>` and `List<TsgcDiagramConnector>` .
UndoStack é um **TObject**.	Continua sendo um `object` : faça o cast para **TsgcDiagramUndoStack**.
Free.	O coletor de lixo. As views WPF e Avalonia são `IDisposable` , chame **Dispose** quando a janela fechar; uma view WinForms é descartada com o seu formulário. **TsgcDiagramController** e **TsgcDiagramDataBinding** são `IDisposable` .
O diagrama transmitido por stream no arquivo DFM ou FMX.	A propriedade oculta **DiagramJSON**, gravada pelo designer do WinForms somente quando o diagrama não está vazio, e utilizável como atributo XAML.
Códigos de tecla e **TShiftState**.	Códigos de tecla virtuais `int` (**CS_DIAGRAM_KEY_***; os valores **Keys** do WinForms são os mesmos) e a enumeração `[Flags]` **TsgcDiagramShiftState**.
As cores são **TsgcUIColor**.	Valores ARGB `uint` , por exemplo `0xFFDBEAFE` ; 0 é a cor do tema.
Exportação PNG do **TsgcDiagramRenderer**.	Métodos de extensão de esegece.sgcDiagram.Rendering, desenhados com o Skia. Os arquivos PNG não são idênticos byte a byte aos do Delphi; o texto SVG e JSON é.
TsgcDiagramDataBinding é um componente com um **TDataSource**.	Uma classe simples com um `DataTable` , veja Vinculação de dados no .NET.

Exemplo

Eventos com parâmetros `ref`, um conjunto de flags e os métodos que substituem propriedades do Delphi:

Comportamento que é o mesmo

- Unidades do mundo, o eixo y para baixo, ângulos em graus no sentido horário, a grade, o encaixe e a ordem de acerto.

- As legendas de desfazer, um passo de desfazer por transação, o formato da área de transferência interna e o payload de arrasto da paleta.
- O formato JSON, as rotas, os layouts e a exportação SVG, com os mesmos números e texto do Delphi.

Veja também

- Visão geral do .NET
- Classes do .NET
- Eventos
- Formato JSON

Vinculação de dados no .NET

O **TsgcDiagramDataBinding** do Delphi constrói um diagrama a partir de um TDataSet: um nó por registro e um conector de cada registro para o seu pai. No .NET a mesma classe lê um `System.Data.DataTable`, a tabela que um `DataAdapter` de qualquer provedor ADO.NET preenche, ou uma **IsgcUITable** do sgcUICore. É um recurso da edição Registered.

Propriedade	Descrição
DataSource	A `DataTable` dos registros.
Table	Uma **IsgcUITable**, lida antes de **DataSource** quando atribuída.
KeyField, ParentField, ParentSeparator, LabelField	A coluna da chave, a coluna da chave do pai (vários pais separados por **ParentSeparator**, um ponto e vírgula por padrão) e a coluna do texto do nó.
Target, Document	**Target** é uma view de diagrama, ou qualquer objeto que implemente **IsgcDiagramDocumentOwner**; **Document** é um documento no qual escrever diretamente, para um programa sem view.
Layout, Direction, TreeStyle	O layout do diagrama reconstruído (**dgbAuto**, **dgbTree**, **dgbLayered**, **dgbNone**) e suas opções.
NodeShape, NodeWidth, NodeHeight, ConnectorKind	A forma e o tamanho dos nós e o tipo dos conectores.
LiveUpdates, Active	O diagrama é reconstruído após uma alteração da tabela enquanto **LiveUpdates** for true; definir **Active** o constrói.
OnCreateNode, OnCreateConnector	Disparados para cada nó e conector criado, com o índice da linha. **CurrentRow** é o `DataRow` do nó enquanto **OnCreateNode** executa.

Exemplo

Diferenças em relação ao TDataSet

- Um `DataTable` não tem cursor, então **OnCreateNode** lê a linha por **CurrentRow** em vez dos campos do dataset.
- O vínculo escuta os eventos da tabela: uma linha adicionada, alterada ou excluída, uma alteração rejeitada e uma tabela limpa reconstroem o diagrama uma vez. `AcceptChanges` e a navegação pelas linhas não reconstroem nada. Uma linha em edição (`BeginEdit`) é usada quando o seu `EndEdit` executa. Linhas excluídas e desanexadas são ignoradas.
- A reconstrução é postada, como o **ForceQueue** do Delphi. Em uma aplicação WinForms, WPF ou Avalonia ela é postada no **SynchronizationContext** atual quando o vínculo foi criado, então nada precisa ser chamado. Um programa de console ou um serviço não tem contexto: chame **CheckQueued** na thread do documento, o **CheckSynchronize** do Delphi, ou **Refresh** para reconstruir imediatamente.
- O vínculo não é um componente: é `IDisposable`, e uma tabela ou destino descartado se desvincula sozinho.

Vincular os controles

As propriedades dos controles são propriedades CLR simples, não propriedades de dependência ou estilizadas, porque um diagrama é um modelo inteiro e não um único valor. Em uma aplicação MVVM mantenha o **TsgcDiagramDocument** ou o seu JSON no view model e reaja a **OnChange** e **OnSelectionChange** da view.

Veja também

- Visão geral do .NET
- Diferenças do .NET em relação ao Delphi
- TsgcDiagramDataBinding
- Vínculo de Dados

Renderização sem janela no .NET

O documento não precisa de janela. Um serviço, um teste ou um relatório pode construir ou carregar um diagrama e salvá-lo como JSON, SVG e PNG em qualquer plataforma, inclusive no Linux.

- **TsgcDiagramDocument** é criado com `new TsgcDiagramDocument()`. Os conectores são roteados quando a transação ou atualização mais externa termina.
- **TsgcDiagramRenderer** grava o diagrama inteiro com uma margem (20 por padrão) em uma escala: **SaveToSVGFile** e **SaveToSVGStream** com esegece.sgcDiagram, **SaveToPNGFile** e **SaveToPNGStream** com esegece.sgcDiagram.Rendering.
- **SaveToJSON**, **LoadFromJSON**, **SaveToFile**, **LoadFromFile**, **SaveToStream** e **LoadFromStream** do documento leem e gravam o formato de arquivo do Delphi.

Exemplo

Layout automático

Na edição Registered um diagrama construído em código pode ser organizado antes de ser salvo. **Execute** é um passo de desfazer e roteia os conectores novamente.

Notas

- O tamanho da imagem é o limite do conteúdo mais a margem, vezes a escala; uma escala de 2 grava uma imagem com o dobro da largura e da altura.
- **Theme** do documento escolhe as cores claras ou escuras da exportação.
- **Painter.SetFontFile** do renderer recebe um arquivo de fonte TrueType regular e um em negrito, para um servidor sem a fonte padrão.
- Os controles também têm **SaveToPNG** e **SaveToSVG**.

Veja também

- Visão geral do .NET
- Classes do .NET
- Início Rápido Sem Janela
- TsgcDiagramRenderer
- Layout em Camadas
- Layout em Árvore

Demos do .NET

As demos estão em `.net\demos\Diagram`, uma pasta por demo com o seu projeto, o seu `Program.cs` e um README.md. Elas constroem a interface em código, sem XAML. Toda demo tem uma opção `--test` que executa uma autoverificação fora da tela, grava linhas PASS e FAIL e termina com o número de verificações que falharam, e uma opção `--snapshot` que grava o diagrama em um arquivo PNG.

Demo	Host	Mostra
FlowchartEditor	WinForms	Uma paleta e uma view, a gêmea da demo FlowchartEditor do Delphi: arraste formas da paleta, conecte-as com a ferramenta de conector ou a partir de uma porta, desfazer e refazer, zoom, temas, o layout em camadas e arquivos JSON, PNG e SVG.
FlowchartWpf	WPF	O mesmo editor com **TsgcWpfDiagramPalette** e **TsgcWpfDiagramView**.
FlowchartAvalonia	Avalonia	O mesmo editor com **TsgcAvaloniaDiagramPalette** e **TsgcAvaloniaDiagramView**, para Windows, Linux e macOS, com zoom por pinça e deslocamento com um dedo.
OrgChart	WinForms	Um `DataTable` de 60 funcionários em um `DataGridView` e, por meio de um **TsgcDiagramDataBinding**, um organograma em uma view somente leitura com o layout em árvore; uma edição na grade atualiza o organograma. Edição Registered.
NodeEditor	WinForms	Um editor de grafos de nós: blocos com portas de entrada e saída tipadas, conectados somente quando os tipos coincidem.
ConsoleRender	Console	Sem janela: constrói um fluxograma de tíquetes de suporte em código, o organiza com o layout em camadas e grava flowchart.png, flowchart.svg e flowchart.json, no Windows e no Linux. A gêmea da demo Headless do Delphi.

A parte sem interface das três demos Flowchart (o fluxograma de exemplo, as ações da barra de ferramentas e a autoverificação) está em `FlowchartEditor_Shared.cs`, que os projetos WinForms, WPF e Avalonia compilam, de modo que os três hosts executam as mesmas verificações.
Na edição Community as demos Flowchart mantêm as posições aproximadas do exemplo e ocultam o botão de layout, ConsoleRender posiciona as linhas em código, e OrgChart mostra a tabela e informa que o organograma exige a edição Registered.

Compilando as demos

Dentro da árvore do produto as demos compilam contra os projetos da biblioteca. O zip das demos usa por padrão os pacotes NuGet da edição Community (OrgChart usa os Registered). Compile uma demo com os pacotes Registered definindo **SgcDiagramEdition**:

```
dotnet build FlowchartEditor\FlowchartEditor.csproj -p:SgcDiagramEdition=Registered
dotnet run --project ConsoleRender\ConsoleRender.csproj -- --test
```

Veja também

- Visão geral do .NET
- Instalação pelo NuGet
- Início rápido WinForms
- Renderização sem janela no .NET

Solução de Problemas

Problema	Causa e solução
Os conectores criados no código não têm rota, ou a rota não acompanha um nó movido.	As rotas são calculadas no fim do par BeginUpdate e EndUpdate mais externo, e somente enquanto Routing.AutoRoute é True. Feche todo BeginUpdate, ou chame RouteAll depois de uma alteração em massa feita com AutoRoute desativado.
Os conectores ortogonais cruzam os nós.	Na edição Community, os conectores dgcOrthogonal são cotovelos simples sem desvio de obstáculos. A edição Registered roteia ao redor dos nós.
Layers.Add gera EsgcDiagramError.	A edição Community permite uma camada. Veja Camadas.
Um nó de imagem mostra uma caixa riscada.	A chave da imagem não está em Images, ou os dados não são um PNG e nenhum manipulador de OnDecodeImage os decodificou.
Excluir uma imagem gera EsgcDiagramError.	Um nó ainda a usa em ImageKey. Altere ou exclua esses nós primeiro.
Um arraste a partir do lado de um nó selecionado o redimensiona em vez de desenhar um conector.	As alças de redimensionamento de um nó selecionado cobrem suas portas laterais. Clique primeiro em espaço vazio, ou defina Tool como dgtConnector.
O editor de texto não abre.	A view é ReadOnly, o nó ou sua camada é Locked, ou um manipulador de OnTextEditing definiu aAllow como False.
Colar não faz nada depois de copiar em outro aplicativo.	A área de transferência do sgcDiagram é interna ao aplicativo; ela não troca dados com outros programas.
Desfazer não faz nada depois de uma alteração feita pelo código.	Uma alteração fora de BeginTransaction e EndTransaction limpa as pilhas de desfazer e refazer. Envolva as alterações do código em uma transação para torná-las desfazíveis.
Não é possível arrastar formas da paleta no iOS ou no Android.	As plataformas móveis não têm serviço de arrastar. Toque em um item da paleta, depois toque na view.
Tab não percorre os itens em uma view FireMonkey.	Um formulário FireMonkey trata Tab antes do controle com foco e move o foco.
Definir o Id de um item gera EsgcDiagramError.	Os Ids são únicos em um documento. Escolha um Id que nenhum outro nó ou conector use.
As classes de layout e de vínculo de dados estão ausentes.	Elas fazem parte da edição Registered. Com a edição Community, sgcDiagram_Layout e sgcDiagram_Data são units vazias.

Veja Também

- Edições
- Plataformas

Index

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